cells

MDPI

Article

A Hybrid Prediction Method for Plant lncRNA-Protein Interaction

Jael Sanyanda Wekesa [1,2], Yushi Luan [3], Ming Chen [4] and Jun Meng [1,*]

1 School of Computer Science and Technology, Dalian University of Technology, Dalian 116023, Liaoning, China; jael@mail.dlut.edu.cn
2 Department of Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi 62000-00200, Kenya
3 School of Bioengineering, Dalian University of Technology, Dalian 116023, Liaoning, China; luanyush@dlut.edu.cn
4 College of Life Sciences, Zhejiang University, Hangzhou 310058, Zhejiang, China; mchen@zju.edu.cn
* Correspondence: mengjun@dlut.edu.cn; Tel.: +86-411-84706003

Received: 26 April 2019; Accepted: 29 May 2019; Published: 30 May 2019

Abstract: Long non-protein-coding RNAs (lncRNAs) identification and analysis are pervasive in transcriptome studies due to their roles in biological processes. In particular, lncRNA-protein interaction has plausible relevance to gene expression regulation and in cellular processes such as pathogen resistance in plants. While lncRNA-protein interaction has been studied in animals, there has yet to be extensive research in plants. In this paper, we propose a novel plant lncRNA-protein interaction prediction method, namely PLRPIM, which combines deep learning and shallow machine learning methods. The selection of an optimal feature subset and subsequent efficient compression are significant challenges for deep learning models. The proposed method adopts *k*-mer and extracts high-level abstraction sequence-based features using stacked sparse autoencoder. Based on the extracted features, the fusion of random forest (RF) and light gradient boosting machine (LGBM) is used to build the prediction model. The performances are evaluated on *Arabidopsis thaliana* and *Zea mays* datasets. Results from experiments demonstrate PLRPIM's superiority compared with other prediction tools on the two datasets. Based on 5-fold cross-validation, we obtain 89.98% and 93.44% accuracy, 0.954 and 0.982 AUC for *Arabidopsis thaliana* and *Zea mays*, respectively. PLRPIM predicts potential lncRNA-protein interaction pairs effectively, which can facilitate lncRNA related research including function prediction.

Keywords: autoencoder; random forest; light gradient boosting machine; hybrid; lncRNA-protein interaction; plant

1. Introduction

Ribonucleic acid (RNA) is a polymeric single-stranded molecule that constitutes living cells alongside deoxyribonucleic acid (DNA) and protein. Long non-coding RNAs (lncRNAs) are broad and myriad group of endogenous single-stranded polynucleotides non-protein coding transcripts with greater than 200 nucleotides sequence length [1]. Previously, lncRNAs were thought to be transcriptional noise because they are less efficiently spliced compared with messenger RNAs (mRNAs). Nevertheless, accumulated experimental evidence has suggested that lncRNAs have indispensable roles in development, hormone-dependent signaling, and stress responses in plants [2,3]. For example, lncRNA COLDAIR in *Arabidopsis thaliana* regulates flowering time and floral organ formation during vernalization [4]. Functional roles of lncRNAs in plants are lagging behind compared with other organisms such as animals, bacteria, and viruses [5]. The mechanism underlying the functions of

lncRNAs is a captivating area of research. More and more studies have indicated that regulation of transcriptional, post-transcription processes and new insights into potential functions can be elucidated by lncRNAs binding with proteins [6]. Additionally, information about these interactions can hint at molecular causes of diseases in plants and animals [7]. lncRNA-protein interactions have been revealed by various experimental methods. For instance, using a fluorescence resonance energy transfer (FRET) approach coupled with fluorescence lifetime imaging microscopy (FLIM) to detect RNA-protein interactions in plant leaves [8]. Other experimental methods such as RNA-protein pull-down assays and RNA immunoprecipitation protocol have been adapted for detecting RNAs and their protein interaction partners in plants [9,10]. However, these wet-lab experimental methods are time-consuming, labor-intensive, and relatively few lncRNA-protein associations have experimental proof. Therefore, computational prediction of RNA and protein interaction partners to complement experimental approaches has become increasingly crucial.

To hasten research, many genome databases have been set up. The databases and web-based platforms available for human and mammals blend lncRNA information including functional annotation, phylogenetic conservation, associations with diseases, and relations between lncRNAs with other RNAs and proteins [5]. However, databases for plant lncRNAs are incomprehensive [5]. Over the last decade, many plant lncRNAs have been discovered by both experimental and computational screening [11]. Computational tools such as LncFinder [12], PlncPRO [13], PlantRNA Sniffer [14], and LncADeep [15] have produced a huge influx of plant lncRNA data. Two main approaches prevalent across lncRNA-protein interaction methods are feature-based and similarity-based [16]. Similarity-based methods assume that lncRNAs with similar functions interact with similar proteins, a concept known as 'guilt-by-association'. Feature-based methods extract features from input data and employ methods such as support vector machine (SVM) to predict interactions between lncRNAs and proteins. In previous research, lncRNA-protein association prediction is based on biological characteristics such as sequence, structure, localization, and genomic context. For example, catRAPID [17] used physicochemical properties including secondary structure, hydrogen bonds, and van der Waals forces between proteins and lncRNAs to assess the tendency of interaction. Lu et al. developed lncPro [18] for lncRNA-protein interaction based on Van der Waals propensities, hydrogen bonding, and secondary structures. RPI-Pred [19] predicted ncRNA-protein interactions by integrating 3D structural and sequence features. LPI-ETSLP [20], a semi-supervised method based on sequence data, revealed lncRNA-protein association using eigenvalue transformation without the need for negative samples. Zhao et al. developed IRWNRLPI [21], a method that integrates random walk and neighborhood regularized logistic matrix factorization for forecasting lncRNA-protein association. Moreover, many network-based methods have been proposed to predict lncRNA-protein interaction based on the integration of heterogeneous networks including LPIHN, RWR, and LPI-NRLMF [22–24].

Traditional methods use hand crafted features that consume more time, require strong domain knowledge, and are problem-dependent. Therefore, it is immensely beneficial to develop a feature extraction method that is multi-layered (deep) to enhance representation power and provide insight into complex features [25]. Deep learning (DL) methods compute high-level representations from raw input. These models have been broadly utilized by researchers to understand the molecular mechanism in human and plant diseases [26–28]. As the hottest sub-field of machine learning, DL methods extract intrinsic features through multilayer architectures such as autoencoders (AE), convolutional neural networks (CNN), and recurrent neural networks (RNN) [29]. In computer vision, speech recognition, and text processing, DL has been observed to perform better than other popular machine learning methods [30]. It has also been successfully implemented in bioinformatics [31–33]. For example, DeepBind [31] is a DL-based model developed to identify sequence specificities of DNA and RNA binding protein. Similarly, DeepSEA [32] predicted noncoding-variant effect from chromatin-profiling sequences using DL. Recently, Pan et al. [34] predicted RNA-protein interaction by local-global deep CNN. Despite their significantly outstanding performance, selection of an optimal feature subset and subsequent efficient compression are still significant challenges in the existing methods. Therefore,

deriving the most salient features is essential for improving performance. This can be achieved by imposing sparsity constraints including non-negativity on weights, weight decay regularization, and adding l_1/l_2-regularization terms to loss function [35,36]. l_1/l_2-regularization terms, referred to as mixed norm, minimizes reconstruction error in the AE model during training. Other constraints associated with AE include randomly corrupted data inputs, referred to as denoising AEs [37].

The most appealing characteristic of DL is the ability to extract high-level features to leverage the large number of unlabeled instances. Based on the extracted features, classifiers such as support vector machine (SVM) and random forest (RF) are used to build the prediction model. However, while a single classification model performs well, fusing multiple models through ensemble learning improves performance. An ensemble learning method is a meta-algorithm that trains several baseline models and combines them into a single predictive model. Apart from improved classification, ensemble methods perform well in problems involving noisy and imbalanced datasets [38]. Classification strengths of individual base classifiers selected for construction of the overall ensemble model lead to more accurate predictive performance. In 2011, a method named RPISeq [39] extracted 3-mer and 4-mer sequence features to train RF and SVM models for prediction of protein-RNA interaction. Then, Wang et al. [40] presented a model that predicted interactions between proteins and RNAs based on Naive Bayes (NB) and an extended NB classifier. In 2016, Pan et al. developed IPMiner, a sequence-based method for predicting lncRNA-protein interactions based on stacked autoencoder [41]. Yi et al. proposed RPI-SAN for lncRNA-protein interaction based on stacked autoencoder and RF [42]. lncLocator [43] predicted lncRNA subcellular localizations based on an ensemble of SVM and RF classifiers. A recent tool termed HLPI-Ensemble (human lncRNA-protein interaction) was proposed to predict human lncRNA-protein interactions based on SVM, extreme gradient boost (XGB), and RF [44]. For cancer prediction, a model based on DL with an ensemble of three classifiers was recently proposed by Xiao et al. [45].

In this study, we present a hybrid learning-based method, termed PLRPIM. Our objective is to predict lncRNA-protein interactions for plant species and demonstrate its significance in lncRNA annotation since most plant lncRNAs functions are unknown. Intrinsic high-level features are extracted by stacked sparse AE, a deep generative model. We use *k*-mer sparse matrices to represent lncRNA and protein sequences. Finally, the extracted features are fed into an ensemble of two classifier models, RF and light gradient boosting machine (LGBM) for prediction. The final outcome is computed by implementing a majority voting mechanism to the individual classification outcomes. The main contributions are: 1) We apply DL to extract high-level abstraction features from lncRNA and protein sequences; 2) adopt dropout, ReLU activation function, and l_1/l_2-regularization sparsity penalties to significantly reduce overfitting and improve performance; 3) fully exploit sequence information from *k*-mer and heterogeneous ensemble strategy to further improve prediction accuracy. Experimental results on two datasets including *Arabidopsis thaliana* and *Zea mays* show that our method performs better than other state-of-the-art methods such as RPISeq-RF, RPI-SAN, and IPMiner.

2. Materials and Methods

2.1. Dataset

We used sequence data from the plant lncRNA database (PlncRNADB) of lncRNA and their RNA-binding protein partners. The lncRNA-protein interaction data of *Arabidopsis thaliana* and *Zea mays* are downloaded from the website: http://bis.zju.edu.cn/PlncRNADB. *Arabidopsis thaliana* has 390 lncRNAs and 163 RNA-binding proteins whereas *Zea mays* has 1107 lncRNAs and 190 RNA-binding proteins. The dataset for *Arabidopsis thaliana* contained 948 positive samples (interactive pairs) and the *Zea mays* dataset contained 22,133 positive samples. The same number of negative pairs (948 and 22,133) were generated through randomly pairing proteins with lncRNAs and further removing the existing positive pairs [39]. Then, we used 20% of the data as the hold-out test set. Details of the datasets are shown in Table 1.

Table 1. Details of lncRNA–protein interaction datasets for two species.

Species	Dataset	Positive Samples	Negative Samples	Total
Arabidopsis thaliana	Training set	758	758	1516
	Test set	190	190	380
Zea mays	Training set	17,706	17,706	35,412
	Test set	4427	4427	8854

2.2. Interaction between LncRNA and Protein

We consider sequence features that are likely to be significant in predicting protein-lncRNA interactions. The complex molecular features include the number of atoms, hydrogen bonds, evolutionary conservation score, mutual interaction propensity, and electrostatic charges [46]. The interaction information between lncRNA and protein is derived from biological knowledge. The positive set consists of a concatenation of lncRNA-protein interacting pair l_i and p_j while the negative set consists of non-interacting pairs. To obtain lncRNA-protein interactions, we denote an interaction matrix Y as $Y(l_i, p_j)$. $Y(l_i, p_j) \in \{0,1\}$ where 1 indicates an interaction between l_i and p_j and 0 indicates no interaction. To define interaction profiles of lncRNA and protein, we use variables $m = SW(l_i, l_j)$ and $n = SW(p_i, p_j)$, where SW refers to Smith-Waterman (SW) algorithm [21]. $SW(l_i, l_j)$ and $SW(p_i, p_j)$ are the measures of similarity between sequence of lncRNA l_i and l_j and sequences of protein p_i and p_j. The underlying presumption of m and n is that lncRNAs collaboratively interact with akin protein partners and vice versa.

2.3. Sequence Feature Encoding

Feature encoding is a key task during construction of a sequence-based machine learning model for prediction. Sequence feature encoding in this work is based on n-gram technique, also known as k-mer of length k where lncRNA is represented by (k_1 = 1, 2, 3, 4) and proteins (k_2 = 1, 2, 3) descriptors. In our experiments, protein and lncRNA sequences are coded using two methods: 1) Autocovariance and 2) conjoint triad. We investigate sequence similarities between lncRNAs and proteins to determine the combination of features. A feature vector of a varied length of size m for lncRNAs and n for proteins is generated. We calculate sequence similarity in lncRNAs using sequence information. Let S_{lncRNA} represent a set of lncRNA sequence of interest. In order to compute similarities between lncRNA l_i and l_j, the following formula is used:

$$S_{\ln cRNA}(l_i, l_j) = \frac{SW(l_i, l_j)}{\max(SW(l_i, l_i), SW(l_j, l_j))} \tag{1}$$

We encode amino acid for protein sequences by natural numbers selected randomly. A feature vector of a varied length of size n is generated. Protein sequences are computationally analyzed by mapping amino acid bases into the numeric values. Let S_{protein} represent a set of the protein sequence of interest. Given proteins i and j, we find the similarity between them based on SW dynamic programming algorithm using the following formula:

$$S_{\text{Protein}}(p_i, p_j) = \frac{SW(p_i, p_j)}{\max(SW(p_i, p_i), SW(p_j, p_j))} \tag{2}$$

2.4. The PLRPIM Pipeline

Machine learning is applied in RNA-protein interaction (RPI) prediction to address the challenge of sparsity of known RPIs. The essence is to select effective feature combinations. In this study, we present a stacked AE framework with sparse constraints for lncRNA-protein interaction prediction, namely PLRPIM. PLRPIM is a neural network model described as follows. First, each known lncRNA-protein interaction pair is assigned 1 and 0 to non-interacting pairs. Secondly, AE maps a pair of input protein

and lncRNA sequences to representations. Then, the integrated lncRNA and protein feature matrices $\{X_i\}_{i=1}^{m}$ and $\{W_i\}_{i=1}^{n}$ are transformed into feature vectors. Feature vector for lncRNA is represented as $F_{lncRNA} = \{F_{l1}, F_{l2}, \ldots, F_{lm}\}$ and $F_{protein} = \{F_{p1}, F_{p2}, \ldots, F_{pn}\}$ for protein. Therefore, two subnetworks of lncRNA and protein are generated. Finally, a softmax layer is used to merge the two separate feature vectors and integrate them into a fully connected neural network to predict the label of each pair. lncRNA-protein interaction matrix M_{ij} generated from predicted lncRNA-protein interaction matrix is the expected output matrix. To improve the performance of the proposed method in terms of robustness and accuracy, we adopt an integration strategy. We merge results from two machine-learning methods, RF and LGBM, to construct the prediction framework. The technical workflow of the proposed method is given in Figure 1.

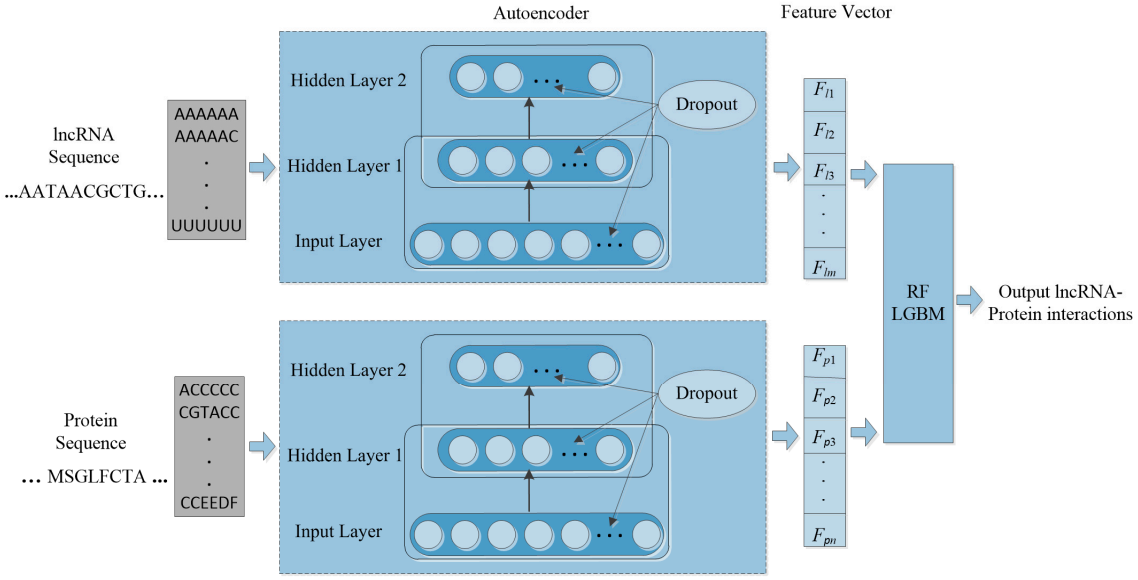

Figure 1. The workflow of PLRPIM model.

2.4.1. Feature Extraction

Features are the sequence-based consolidated attributes of lncRNAs and proteins encoded into numeric vectors that are used in the prediction. In this work, we selected the *k*-mer model to extract features from lncRNA and protein, where the length of genetic sequence subset *S* is represented by an integer *k*. The *k*-mer method has been successfully used in extracting features from RNAs and amino acid composition of proteins based on local sequence patterns. RNA sequence consists of symbols from four alphabets, A, G, C, and T. RNA sequences are represented as the occurrence frequencies of *k*-mers following the principle of complementary pairing [47]. Each *k*-mer pattern is represented as 4^k for each *k* value, where 4 is the number of RNA sequence nucleotides. For example, 16 patterns (e.g., "AA","AC","AT", ...) can be obtained when $k = 2$. We extracted 4^4 number of lncRNA sequence features by utilizing the *k*-mer scheme. Each lncRNA sequence is converted into a *k*-mer sparse matrix from which feature vectors are extracted. A protein chain has amino acids with 20 alphabets (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y).

To obtain highly efficient features, we extracted a set of 599 descriptors from feature vectors encoded by various properties of lncRNAs and proteins. A total of 256 features are obtained from lncRNA sequence and 343 amino acid descriptors from protein sequence. We extract 4-mer sparse matrix of RNA sequence (A, C, G, T) by searching each sequence from left to right and obtain 256 ($4 \times 4 \times 4 \times 4$) feature map. For the protein sequences, we partition amino acid compositions based on their chemical similarity. Seven groups of physicochemical properties {Val, Gly, Ala}, {Phe, Pro, Leu, Ile}, {Ser, Tyr, Met, Thr}, {His, Asn, Tpr, Gln}, {Arg, Lys}, {Glu, Asp}, and {Cys} of protein sequences were assigned numbers based on dipole moments (<1.0, <1.0, (1.0, 2.0), (2.0, 3.0), >3.0, >3.0, and <1.0)

and their chain volumes (<50, >50, >50, >50, >50, >50, and <50) [19,41]. We extracted 3-mer counts of group labels to create a sparse matrix of 343 ($7 \times 7 \times 7$) feature map.

2.4.2. Hybrid Learning Method

DL architectures capture complex correlations in feature representations for diverse prediction tasks. The utmost significant property is learning features at multiple abstraction levels and feature concatenation within and across datasets. The core idea of stacked AE via sparse representations is to effectively obtain descriptors of data as linear projections that maximizes the correlation between features. AE is a kind of artificial neural network designed to extract and generate abstract features of high-dimensional data. It applies unsupervised learning algorithm to construct hidden structures from unlabeled data. The process of AE training consists of two components, an encoder and decoder. The encoder (function f) is used for mapping the input data (x, y) into latent representation and the decoder (function g) maps the encoded features to reconstruct input data from the latent representation. In this work, we formulate lncRNA-protein interaction prediction as a binary classification problem. We used constrained stacked AE (CSAE) network for DL and classification of training dataset as shown in Figure 2. The sparsity constraints extract the most informative features for optimal sample selection. Stacked AEs have greater expressive power and the successive layers of representations capture a hierarchical grouping of the input similar to convolution and pooling operations in CNN. Notations $W = \{W^1, W^2, \ldots, W^{nl}\}$ and $b = \{b^1, b^2, \ldots, b^{nl}\}$ denote the AE parameters. Weight matrix W denoted as W_{ij}^l is the weight associated with the connection between neuron unit j in layer l-1 and neuron unit i in layer l, where $j = 1,2, \ldots, sl - 1$, $i = 1,2, \ldots, sl$ and $l = 2,3, \ldots, nl$. sl - 1 represents the output of the previous layer while nl represents the number of layers in the network. Bias vector is denoted as b_i^l where $i = 1, 2, \ldots, sl$ and $l = 2, 3, \ldots, nl$ denotes the bias of neuron unit i in layer l. The training set is defined as $\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$ of n data samples. Suppose x is the input data of dimension d, the AE maps x to y as shown in the following formula:

$$y = f(Wx + b) \tag{3}$$

where f is the encoding function and Wx is a weight matrix that maps the output of x to a hidden-space. After mapping x to y, the output of the encoder (y) is mapped back to form z, which is the transformation of x with the same shape as x. The reconstruction is formulated as follow:

$$z = g(W^T y + b') \tag{4}$$

where g is the decoding function, non-linear function W^T is the weighting matrix for transformation, and b' is the transformation bias. The transformation error is estimated using squared error between x and z. To minimize and optimize the reconstruction error, parameters w and b are adjusted using stochastic gradient descent (SGD) [48]. Cross-entropy, the most common type of loss function, is used to minimize reconstruction error by AE. We use 256, 128, and 64 layers that are fully connected. We model the neurons output $f(x)$ using rectified linear unit (ReLU) activation function:

$$f(x) = \max(x, 0). \tag{5}$$

We trained the AEs by adding dropout layers. Dropout layers regularize the model to avoid the risk of overfitting by randomly leaving out some neuron units.

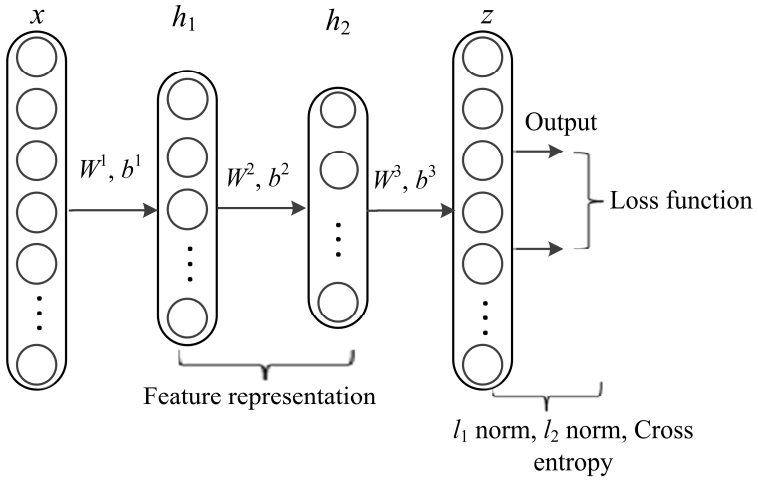

Figure 2. The network structure of constrained stacked autoencoders (AE).

2.4.3. Training of PLRPIM Model with Mixed Norm Constraint

Inducing sparse constraints in AEs including weight decay and norm penalties has been successfully implemented to achieve high discriminative power of models. Inspired by [35,36], we implement mixed norm regularizer constraints to train stacked AE. The sparse l_1 and l_2 penalties for the weights are defined as follows:

$$l_1 = \frac{\lambda}{2}\sum_{i=1}^{m} \|W_{i,j}^{l}\|, \tag{6}$$

$$l_2 = \frac{\lambda}{2}\sum_{i=1}^{m} \|W_{i,j}^{l}\|^2 \tag{7}$$

where $W_{i,j}^{l}$ denotes the connection between neuron unit i and j in layer l.

The proposed model is multi-layered and is trained in three phases. We use feed forward to optimize weights of the networks and adopt backpropagation in the last phase of training the network. The stacked AE-based generated networks are pre-trained before using backpropagation algorithm. The layers in the model include an input layer x with data points $\{(x^1,y^1), (x^2,y^2), \ldots, (x^n,y^n)\}$, hidden layers $\{h_1, h_2\}$, and an output layer z (Figure 2). The features obtained from the hidden layers are used to reconstruct the input data and train the network. The first phase of the training process (pre-training) is realized by minimizing an objective function using cross-entropy (C) to avoid overfitting using Formulas (8) and (9) as follows:

$$C = -(y \log z + (1-y)\log(1-z)) \tag{8}$$

$$\min_{w,b}\left[\sum_{i=1}^{m}(z-x)^2 + \beta\sum_{j=1}^{k} C\right] \tag{9}$$

where y are the labels of training examples, z represents the predicted output from input data x, m is the index of the number of training examples, and k is the index of the number of features. β is the trade-off parameter used to control overfitting. The second phase is made up of two hidden layers (h_1 and h_2). The third phase is the output layer, which receives input from the second hidden layer. In this phase, first, the weight and bias parameters (W^3 and b^3) are initialized. Then, a backpropagation algorithm is adapted to compute derivatives of the objective function and tune weights.

We apply mixed norm regularizer (l_1/l_2) with sparsity coefficient λ to minimize reconstruction error. The mixed norm regularizer is based on the assumption that descriptors from a set of training

examples exhibit similar sparsity patterns in the feature map [49]. Formula (10) shows the computation of the mixed norm regularizer:

$$\min_{w,b}\left[\sum_{i=1}^{m}\|x-z\|^2+\beta\sum_{j=1}^{k}C+\frac{\lambda}{2}\|W\|_{1,2}\right]. \quad (10)$$

The parameter β is used for regularization, $\lambda \geq 0$ is the weights sparsity term, $\|W\|_{1,2}$ represent l_1 and l_2 norms defined in Formulas (6) and (7) where $\|W\|_1 = \sum_{i=1}^{m}\|W_{ij}^{l}\|$ and $\|W\|_2 = \sum_{i=1}^{m}\|W_{ij}^{l}\|^2$. Formula (10) minimizes the average reconstruction error, increase sparsity of latent layer activation, and reduce the number of non-negative weights.

2.4.4. Shallow Classification Models

A vast number of classical machine learning methods have been employed in research and real-world systems to solve classification and regression problems. For RNA-protein interaction problem, SVM and RF shallow classification methods have previously been used [41]. Therefore, we consider these two and four other methods, adaptive boosting (Adaboost), XGB, LGBM, and decision tree (DT) to compare the performance of the single with merged classifiers. We assessed the prediction performance of the six commonly used classification methods. For most artificial intelligence applications, all the classification models are of high accuracy. However, each method may outperform others in different cases. We exemplify determining the most relevant kernel (radial basis function, linear, and polynomial) function challenge for SVM, while RF and gradient boosting decision tree (GBDT) may produce better classification results for the categories with more samples [45]. Therefore, a method that takes advantage of more than one classifier leads to superior prediction performance in practice.

The most widely used method for RNA-protein interaction prediction is SVM [50]. SVM is non-probabilistic classifier that maps input data points into a high-dimensional feature space. Other methods are ensemble methods, which are categorized into three classes: Bagging, boosting, and stacking. They combine several learning methods to obtain a predictive model with improved performance. They include RF, Adaboost, XGB, and LGBM. Given n models, f_i is averaged into an ensemble e:

$$e(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x). \quad (11)$$

RF is a widely used shallow machine learning method that combine decision tree predictors following the bagging technique [45]. In this model, the class that receives majority votes from trees in the forest is considered the output result. This protocol relies on creating n number of models and averaging predictions of all models for a final prediction. In this work, the RF contained 50 decision trees with a minimum leaf size of 3 for each tree.

Adaptive boosting (AdaBoost) is an ensemble method often used to obtain satisfactory results compared with other methods [38]. It aims at converting a set of weak classifiers into a strong one. In this work, AdaBoost included 50 shallow decision trees.

GBDT is a machine learning technique comprised of a collection of decision trees to form a stronger prediction model. GBDT builds the model in a stage-wise fashion and trains it iteratively. It implements generalization by allowing optimization of an arbitrary differentiable loss function, which makes them efficient, accurate, and interpretable. Light gradient boosting (LGBM) is a tree-based learning method that implements GBDT [51]. It is suitable for the large size of data and a large number of features. It trains fast, utilizes low memory, and its accuracy is better. Unlike other methods that follow level-wise training pattern, LGBM and XGB follow a leaf-wise training approach [52].

XGB is an advanced GBDT method designed for speed, flexibility, and accuracy [53]. This tree ensemble model is trained in an adaptive manner. It has been used to build a prediction model for protein-protein interface residues [54]. In this work, the features for XGB were selected by its internal algorithm. Features for the classification models were selected by a backward elimination

manner. The labels used by the classification methods for prediction were determined by stacked AE. The classification models were then used to test accuracy and robustness in test sets. Since our dataset is a pair-wise sample, the feature combinations with the highest predictive accuracy were selected and fed into an ensemble of the two models to evaluate the performance. Five-fold cross validation technique was used to evaluate the predictive ability of the proposed method. A hybrid method was employed, which combined RF and LGBM classifiers. These classifiers cannot directly work using raw sequence data. Therefore, an unsupervised DL model is implemented for extraction of features. We employ majority voting, a widely used method for the fusion of multiple classifiers.

2.4.5. Experimental Setup

In this study, we used *Arabidopsis thaliana* and *Zea mays* sequence datasets. The source codes for the experiments were written in python 2.7 programming language. For each dataset, we train our models using Keras library (https://github.com/fchollet/keras) with Tensorflow [55] as the backend. We separate the data in the ratio 4:1. Each time, four folds are used as the training set and one is withheld as the test set. We feed the training set into our models and use 20% as a test set. During the training process, test data are used to monitor convergence at the end of each epoch. Adding the number of neurons per layer indicates increasing the number of feature maps. In this experiment, we fix the depth of network to 3. We stack a combination of batch normalization, ReLU layer, and dropout layer to form the encoder. We use three different values for the learning rate (0.5, 1, 2), as shown in Table 2, and the dropping probability 0.6. The momentum update method [56] was employed with a momentum coefficient of 0.9. During the training, we set the maximum number of iterations to 50 for all the training examples. For each iteration, training examples are randomly shuffled to speed up the training process. We set a mini-batch training with a batch size of 64 ($m = 64$).

The model is fine-tuned by back-propagation using categorical cross-entropy loss function, which is optimized by SGD with momentum. Other hyper-parameter values that were optimized during training were the learning decay rate (1e-6) and a mean squared error (MSE). Mixed-norm constraints ($l_1 = 0.01$ and $l_2 = 0.01$) are used to minimize reconstruction error. We exploit non-linear activation function on hidden neurons to combat nonlinearity in lncRNA-protein interactions. Encoder function uses ReLU activation function while the decoder function uses sigmoid activation function.

Additionally, we employ batch normalization and early stopping regularization techniques to avoid overfitting. More details on parameter settings for our model are shown in Table 2. We observe that initially, the performance of PLRPIM increases with the depth of the network. However, the performance drops when the depth of the network is greater than three. This is due to overfitting since increasing the number of hidden layers decreases the training data. Figure 3 summarizes the steps followed during the testing of the proposed method.

Table 2. Hyper parameter settings.

Name	Settings
Learning rate	0.5, 1, 2
Parameter optimization	SGD, momentum, Adam
Batch size	256, 128, 64
Activation	ReLU
Loss	MSE, Cross Entropy
Dropout rate	0.6

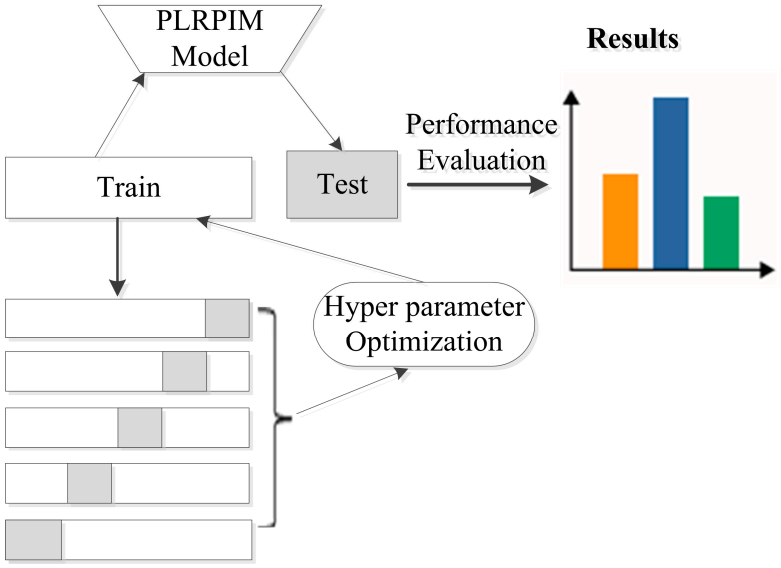

Figure 3. Experimental setup for testing the proposed method. The datasets are split into five folds and hyper parameters are tuned in the training set. The model is learned and applied on the test set based on the hyper parameters. Performance metrics are calculated for all folds.

2.4.6. PLRPIM Method

PLRPI predicts based on the harmonious blend of information from concatenation fusion of sparse representations of lncRNA and protein sequences. The pseudocode of the proposed method is presented as follows (Algorithm 1).

Algorithm 1: Pseudo code of **PLRPIM**

Input: Y—adjacency matrix of interacting lncRNA i and protein j; ($Y(l_i, p_j)$), S_{lncRNA}—set of lncRNA sequence of interest, S_{protein}—set of protein sequence of interest, number of stacked AEs = T, number of iterations (epoch) = R

Output: lncRNA-protein interaction matrix M_{ij}

1. B = {1, 0}—Binary domain;
2. Interact: $S_{\text{lncRNA}} \times S_{\text{protein}} \rightarrow$ B;
3. Interact: ($S(l_i)$, $S(p_i)$)- check whether lncRNA l_i and protein p_i interact; if true, return 1; otherwise, return 0;
4. Concatenate: $S_{\text{lncRNA}} \times S_{\text{protein}} \rightarrow Y$;
5. Initialize training examples labels ($Y(l_i, p_j)$) = 0;
6. **For** t = 1 to T **do;**
7. **For** r = 1 to R **do;**
8. Minimize objective function using formula (10);
9. Compute number of features from similarity matrices $\{X_i\}_{i=1}^{m}$ and $\{W_i\}_{i=1}^{n}$;
10. Compute hidden vector $\{Z_i\}_{n=1}^{N}$ with learned AE;
11. Update $\{Z_i\}_{n=1}^{N}$;
12. **End for;**
13. Compute feature vectors $\{F_{l1}, F_{l2}, \ldots, F_{lm}\}$ and $\{F_{p1}, F_{p2}, \ldots, F_{pn}\}$;
14. Predict class labels of the test dataset based on ensemble voting process using formula (11);
15. Update training examples labels ($Y(l_i, p_j)$) = 1;
16. **End for;**
17. **Return:** $M_{ij.}$

2.5. Evaluation of PLRPIM

To empirically assess the prediction capability of the proposed model, we adopt 5-fold cross-validation as well as some commonly used measures. Cross-validation iteratively partitions the dataset into training and test sets to reduce sampling bias. The data were randomly split into five sets of equal size. One fold is withheld as the test set and the remaining four are used as the training set. We use six standard metrics including area under ROC curve (AUC), precision (PRE), sensitivity (SEN), specificity (SPEC), accuracy (ACC), and Mathews correlation coefficient (MCC) to evaluate the average performance. The probability of assigning a high rank to a randomly chosen positive instance over a negative one is plotted on the AUC curve. The greater the value of AUC, the better the predictive power of the model. The outcomes from a classifier can be represented as a confusion matrix. The confusion matrix contains four categories of outcomes, true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP are actual lncRNA-protein pairs that are predicted correctly, TN are the pairs correctly extracted from the negative samples, FP are negative entities falsely predicted, whereas FN are the cases where actual lncRNA-protein pairs are predicted as non-interacting.

$$PRE = \frac{TP}{TP + FP} \tag{12}$$

$$SEN = \frac{TP}{TP + FN} \tag{13}$$

$$SPE = \frac{TN}{TN + FP} \tag{14}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{16}$$

3. Results

In this section, we present 5-fold cross-validation results of PLRPIM on two datasets. Table 3 shows the results obtained from each of the five folds, the average value, and the standard deviation.

Table 3. 5-fold results of the proposed method.

Dataset	Test Set	ACC	PRE	SEN	SPE	MCC	AUC
Arabidopsis thaliana	1	0.9000	0.9250	0.8506	0.9417	0.7995	0.9527
	2	0.8865	0.9368	0.8359	0.9402	0.7784	0.9488
	3	0.8971	0.9344	0.8636	0.9337	0.7970	0.9590
	4	0.9050	0.9353	0.8641	0.9436	0.8117	0.9490
	5	0.9103	0.9223	0.9036	0.9176	0.8206	0.9615
	Average	0.8998 ± 0.008	0.9308 ± 0.006	0.8636 ± 0.02	0.9354 ± 0.009	0.8015 ± 0.01	0.9542 ± 0.005
Zea mays	1	0.9372	0.9394	0.9338	0.9405	0.8744	0.9849
	2	0.9340	0.9380	0.9313	0.9369	0.8681	0.9846
	3	0.9312	0.9318	0.9303	0.9321	0.8624	0.9817
	4	0.9310	0.9350	0.9259	0.9361	0.8620	0.9812
	5	0.9387	0.9367	0.9408	0.9366	0.8773	0.9833
	Average	0.9344 ± 0.003	0.9362 ± 0.003	0.9324 ± 0.005	0.9364 ± 0.003	0.8689 ± 0.006	0.9831 ± 0.001

The proposed method has the highest AUC followed by specificity, also referred to as true negative rate. This indicates that our method identifies the percentage of actual non-interactive pairs more accurately for both datasets. Additionally, the standard deviation values showing the variation in the values obtained by each fold are close to zero, indicating low-variance. This shows the uniformity in the results, thus evidencing that our method has reliable performance.

3.1. Comparison between Classification Models

We applied six classification methods individually including SVM, AdaBoost, DT, XGB, LGBM, DT, and RF. Then, we presented the average prediction results obtained from 5-fold cross-validation technique. SVM does not perform well compared with the other methods in our experiments. Therefore, we do not include it in the results presented in Table 4. The strength of our pipeline is based on a heterogeneous ensemble strategy, which effectively reduces the false positive rate and achieves better performance. Integration of baseline models trained by different high-level feature combinations boosted the performance.

We compare our multi-model ensemble method with other classification methods, as presented in Table 4. Results obtained indicate that the proposed method increases prediction accuracy for the datasets compared with single classifiers. From the results presented in Table 4, our method performed better in terms of accuracy, precision, specificity, MCC, and AUC for the *Arabidopsis thaliana* dataset. LGBM had the highest sensitivity. For the *Zea mays* dataset, PLRPIM had the highest accuracy, precision, sensitivity, specificity, MCC, and AUC.

Table 4. Predictive performance of classifiers and the proposed method.

Dataset	Method	ACC	PRE	SEN	SPE	MCC	AUC
Arabidopsis thaliana	PLRPIM	0.8998	0.9308	0.8636	0.9354	0.8015	0.9542
	LGBM	0.8950	0.9182	0.8668	0.9230	0.7912	0.8949
	XGB	0.7452	0.7615	0.7661	0.7187	0.4993	0.8352
	RF	0.7088	0.6851	0.8164	0.5951	0.4294	0.8171
	AdaBoost	0.6962	0.6829	0.8234	0.5622	0.4214	0.8061
	DT	0.6233	0.6331	0.6060	0.6359	0.2478	0.6284
Zea mays	PLRPIM	0.9344	0.9362	0.9324	0.9364	0.8688	0.9831
	LGBM	0.9317	0.9331	0.9300	0.9333	0.8634	0.9317
	XGB	0.7936	0.7689	0.8426	0.7446	0.5909	0.8862
	AdaBoost	0.7849	0.7676	0.8182	0.7516	0.5725	0.8693
	RF	0.7536	0.7407	0.7972	0.7103	0.5111	0.8641
	DT	0.6523	0.6500	0.6676	0.6368	0.3049	0.6894

Figure 4a shows that our method had better performance in terms of AUC on the *Arabidopsis thaliana* dataset. In comparison to the other methods, our method had approximately 15% better performance. The contribution of the two models integrated in PLRPIM significantly boosted the performance. Figure 4b shows that our method had better performance in terms of AUC on the *Zea mays* dataset. In comparison to the other methods, our method had approximately 13% better performance. The performance of all the methods increased in terms of AUC in the *Zea mays* dataset due to the substantial increase in the size of the dataset.

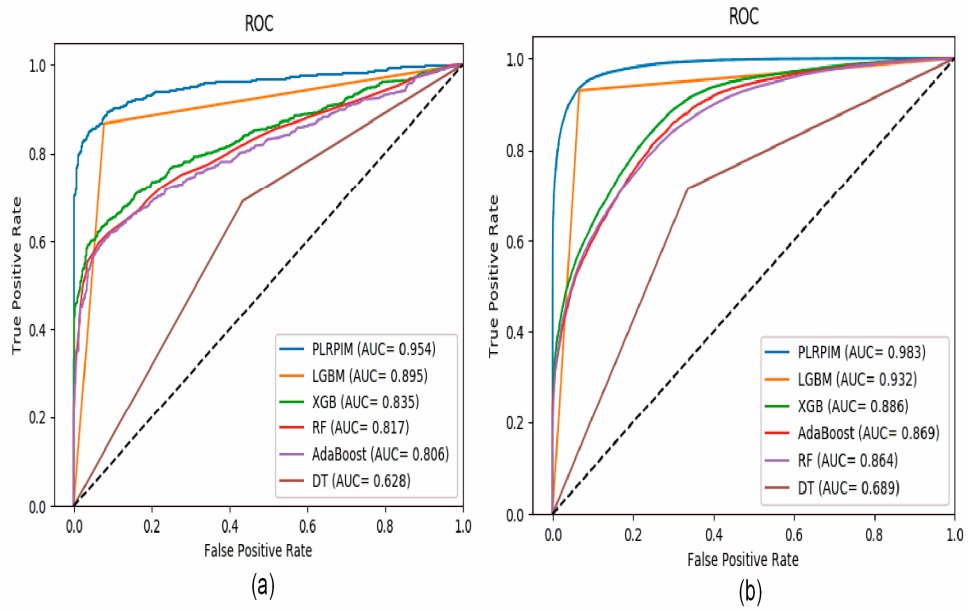

Figure 4. ROC curves for PLRPIM, light gradient boosting machine (LGBM), extreme gradient boost (XGB), random forest (RF), adaptive boosting (AdaBoost), and decision tree (DT) on (**a**) *Arabidopsis thaliana* dataset and (**b**) *Zea mays* dataset.

3.2. Comparison of PLRPIM with Other Existing Tools

To test the reliability of the proposed method, we compare PLRPIM with three other sequence-based computational models, RPISeq-RF, RPI-SAN, and IPMiner. The performance of each tool in terms of accuracy, precision, sensitivity, specificity, and AUC was compared. In terms of accuracy, PLRPIM performed better with 89.98% and 93.44% accuracy for the two plants. The AUC values of PLRPIM, IPMiner, RPISeq-RF, and RPI-SAN for *Arabidopsis thaliana* are 0.9546, 0.8823, 0.8761, and 0.8164, respectively, as shown in Figure 5a. For the *Zea mays* dataset, the AUC values are 0.9823, 0.9034, 0.8980, and 0.8792, respectively, as shown in Figure 5b. Table 5 shows the performance of the four methods.

Table 5. Predictive performance of other methods and the proposed method.

Dataset	Method	ACC	PRE	SEN	SPE	MCC	AUC
Arabidopsis thaliana	PLRPIM	0.8998	0.9308	0.8636	0.9354	0.8015	0.9546
	IPMiner	0.8275	0.8930	0.7448	0.9107	0.6646	0.8823
	RPISeq-RF	0.8059	0.8144	0.7922	0.8200	0.6124	0.8761
	RPI-SAN	0.7579	0.7955	0.6966	0.8199	0.5210	0.8164
Zea mays	PLRPIM	0.9344	0.9362	0.9324	0.9364	0.8688	0.9823
	IPMiner	0.8127	0.8142	0.8106	0.8148	0.6258	0.9034
	RPISeq-RF	0.8069	0.7993	0.8192	0.7945	0.6142	0.8980
	RPI-SAN	0.7890	0.7909	0.7869	0.7911	0.5784	0.8792

By tapping on the performance benefits of sparsity constraints, our model learns the most informative sequence features. In Table 5, our method performs better that the other methods in terms of accuracy, precision, sensitivity, specificity, MCC, and AUC for both *Arabidopsis thaliana* and *Zea mays* datasets.

Figure 5a shows that our method has better performance in terms of AUC on the *Arabidopsis thaliana* dataset. In comparison to the other methods, our method had approximately 9% better performance in terms of AUC. Figure 5b shows that our method has better performance in terms of AUC on the *Zea mays* dataset. In comparison to the other methods, our method has approximately 8% increase in AUC. Increasing the amount of data positively affects the performance of the model.

The better performance is attributed to data size and employing LGBM. LGBM model boosts prediction power since it is suitable for large size of data. All methods are positively influenced by the increase in data size. Their performance in terms of AUC increase.

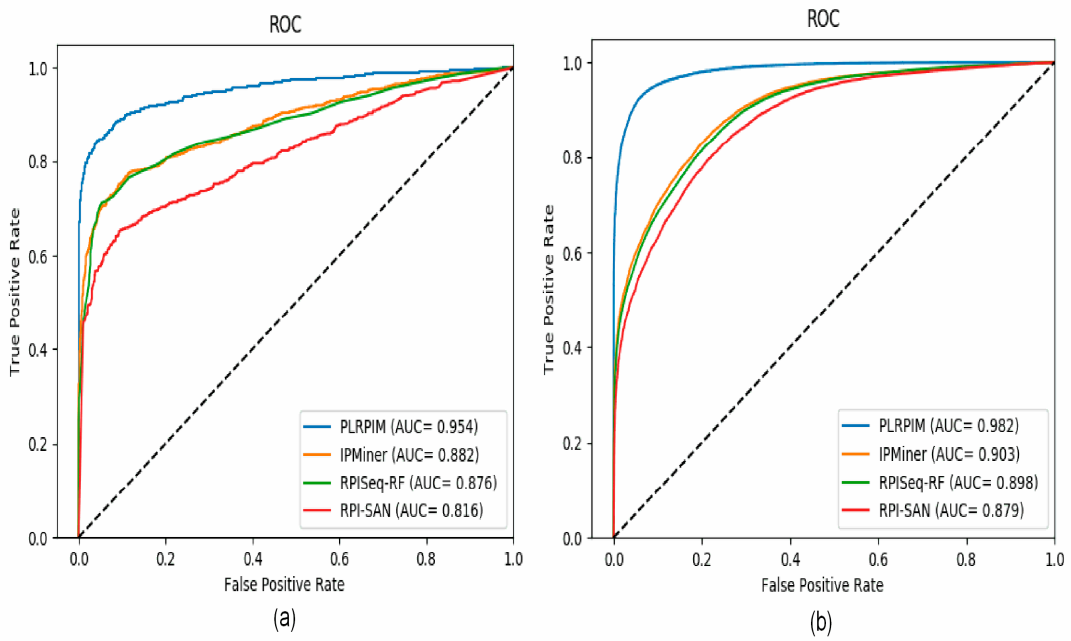

Figure 5. ROC curves for PLRPIM, IPMiner, RPISeq-RF, and RPI-SAN on (**a**) *Arabidopsis thaliana* and (**b**) *Zea mays* datasets.

3.3. Functional Analysis of lncRNAs

We explore *Arabidopsis thaliana* and *Zea mays* lncRNA-protein networks for annotation of lncRNAs. For the *Arabidopsis thaliana*, TCONS_00011717, TCONS_00008833, and TCONS_00012080 interact with proteins IPR012677 and P41377. IPR012677 functions include oxidoreductase activity, nucleotide binding, and nucleotide acid binding. P41377 functions include ATP binding, mRNA binding, and helicase activity. In our *Zea mays* dataset, some genes lack annotations and are referred to as "orphan genes". According to Arendsee et al., orphan genes are defined as genes whose coding sequences are specific to species [57]. Many orphan genes functions are unknown due to lack of homologs and their uniqueness to specific species. Therefore, identifying their protein interaction partners is an alternative way of annotating them. Some orphan genes without functions include GRMZM5G870099, GRMZM2G562000, GRMZM2G046326, GRMZM5G849473, GRMZM2G107204, and GRMZM2G375531.

As an illustration, we provide Table 6 to show the significance of the interaction between lncRNAs and proteins in plant development and regulation of gene expression. For example, the above-mentioned orphan genes and GRMZM2G374777 (orphan gene), GRMZM2G097084 (orphan gene), GRMZM2G078523, GRMZM2G543070, and GRMZM2G147020 interact with protein B6SIF0. This protein has abiotic stress-related functions such as response to cold, response to water deprivation, and response to osmotic stress. We obtain GO annotation terms from EnsemblPlants database (https://plants.ensembl.org) and maize genetics and genomics database (https://maizegdb.org). We observe that the lncRNAs interacting with the same protein share common functions such as two-leaf expansion stage as shown in Table 6.

Table 6. Some selected *Zea mays* and *Arabidopsis thaliana* long non-coding RNAs (lncRNAs) and their biological functions.

Species	lncRNAs	Biological Functions
Arabidopsis thaliana	TCONS_00011717	GO:0006913—Nucleocytoplasmic transport
	TCONS_00008833	GO:0006083—Acetate metabolic process
	TCONS_00012080	GO:0009867—Regulation of jasmonic acid mediated signaling pathway
Zea mays	GRMZM2G374777	PO:0001052—2 leaf expansion stage PO:0006339—juvenile vascular leaf PO:0009089—endosperm
	GRMZM2G097084	GO:0005524—ATP binding GO:0006183—GTP biosynthetic process GO:0006952—defense response
	GRMZM2G078523	PO:0001052—2 leaf expansion stage PO:0009084—pericarp PO:0009089—endosperm
	GRMZM2G543070	PO:0001052—2 leaf expansion stage PO:0001095—true leaf formation stage PO:0007016—4 flowering stage
	GRMZM2G147020	PO:0001052—2 leaf expansion stage PO:0001095—true leaf formation stage PO:0007016—4 flowering stage PO:0001009—D pollen mother cell meiosis stage

4. Conclusions

In this paper, we presented PLRPIM, a hybrid method for the prediction of plant lncRNA and protein interactions. The proposed method consists of CSAE for feature selection and reduction of feature space dimension, a feed forward, and ensemble classifiers for prediction. The prediction performances demonstrate its efficiency in comparison to other methods. From our results, the hybrid of heterogeneous ensemble classifiers and unsupervised learning implemented by the proposed method performs well without dataset size limitation. By fully exploiting multiple classifiers, the proposed method is shown to have a high success rate for lncRNA-protein interaction prediction based on genome sequence. Additionally, the proposed method is versatile and can be used to predict lncRNA-protein interaction for other plants. However, the proposed method has several potential limitations that need to be addressed in the future. First, the degree of research for plant lncRNA-related proteins for different plant species is limited; thus, known lncRNA-binding protein partners are sparse. Information bias can mislead the measurement of interaction probability between lncRNAs and proteins in plants. More data sources with experimentally validated data can potentially improve the performance further.

Author Contributions: J.S.W. and J.M. designed the methods and conceived the study. J.S.W. and J.M. implemented the entire method and performed the experiments. M.C. collected datasets and revised the manuscript and comments. J.M., M.C., and Y.L. analyzed the prediction results. J.S.W. and J.M. wrote the manuscript. All authors read and approved the final manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (Nos. 61872055, 31872116).

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

References

1. D'Aniello, S.; Spagnuolo, A.; Ceccarelli, M.; Cerulo, L.; Ventola, G.M.M.; Noviello, T.M.R.; D'Aniello, S. Identification of long non-coding transcripts with feature selection: a comparative study. *BMC Bioinform.* **2017**, *18*, 187.
2. Lu, X.; Chen, X.; Mu, M.; Wang, J.; Wang, X.; Wang, D.; Yin, Z.; Fan, W.; Wang, S.; Guo, L.; Ye, W. Genome-Wide Analysis of Long Noncoding RNAs and Their Responses to Drought Stress in Cotton (Gossypium hirsutum L.). *PLoS ONE* **2016**, *11*, e0156723. [CrossRef]
3. Nejat, N.; Mantri, N. Emerging roles of long non-coding RNAs in plant response to biotic and abiotic stresses. *Crit. Rev. Biotechnol.* **2018**, *38*, 93–105. [CrossRef]
4. Kim, D.H.; Xi, Y.; Sung, S. Modular function of long noncoding RNA, COLDAIR, in the vernalization response. *PLOS Genet.* **2017**, *13*, e1006939. [CrossRef]
5. Bhatia, G.; Goyal, N.; Sharma, S.; Upadhyay, S.K.; Singh, K. Present Scenario of Long Non-Coding RNAs in Plants. *Non-Coding RNA* **2017**, *3*, 16. [CrossRef] [PubMed]
6. Kashi, K.; Henderson, L.; Bonetti, A.; Carninci, P. Discovery and functional analysis of lncRNAs: Methodologies to investigate an uncharacterized transcriptome. *Biochim. et Biophys. Acta (BBA)-Gene Regul. Mech.* **2016**, *1859*, 3–15. [CrossRef] [PubMed]
7. Xu, Y.; Wu, W.; Han, Q.; Wang, Y.; Li, C.; Zhang, P.; Xu, H. New Insights into the Interplay between Non-Coding RNAs and RNA-Binding Protein HnRNPK in Regulating Cellular Functions. *Cells* **2019**, *8*, 62. [CrossRef]
8. Camborde, L.; Jauneau, A.; Brière, C.; Deslandes, L.; Dumas, B.; Gaulin, E. Detection of nucleic acid-protein interactions in plant leaves using fluorescence lifetime imaging microscopy. *Nat. Protoc.* **2017**, *12*, 1933–1950. [CrossRef]
9. Bierhoff, H. Analysis of lncRNA-Protein Interactions by RNA-Protein Pull-Down Assays and RNA Immunoprecipitation (RIP). *Cell. Quiescence* **2018**, *1686*, 241–250.
10. Mermaz, B.; Liu, F.; Song, J. RNA Immunoprecipitation Protocol to Identify Protein-RNA Interactions in Arabidopsis thaliana. *Plant Chromatin Dyn.* **2018**, *1675*, 331–343.
11. Liu, X.; Hao, L.; Li, D.; Zhu, L.; Hu, S. Long non-coding RNAs and their biological roles in plants. *Genom. Proteom. Bioinform.* **2015**, *13*, 137–147. [CrossRef]
12. Han, S.; Liang, Y.; Ma, Q.; Xu, Y.; Zhang, Y.; Du, W.; Wang, C.; Li, Y. LncFinder: An integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Briefings Bioinform.* **2018**. [CrossRef]
13. Singh, U.; Khemka, N.; Rajkumar, M.S.; Garg, R.; Jain, M. PLncPRO for prediction of long non-coding RNAs (lncRNAs) in plants and its application for discovery of abiotic stress-responsive lncRNAs in rice and chickpea. *Nucleic Acids Res.* **2017**, *45*, e183. [CrossRef]
14. Vieira, L.M.; Grativol, C.; Thiebaut, F.; Carvalho, T.G.; Hardoim, P.R.; Hemerly, A.; Lifschitz, S.; Ferreira, P.C.G.; Walter, M.E.M.T. PlantRNA_Sniffer: A SVM-Based Workflow to Predict Long Intergenic Non-Coding RNAs in Plants. *Non-Coding RNA* **2017**, *3*, 11. [CrossRef]
15. Yang, C.; Yang, L.; Zhou, M.; Xie, H.; Zhang, C.; Wang, M.D.; Zhu, H. LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics* **2018**, *34*, 3825–3834. [CrossRef]
16. Zhang, W.; Yue, X.; Tang, G.; Wu, W.; Huang, F.; Zhang, X. SFPEL-LPI: Sequence-based feature projection ensemble learning for predicting LncRNA-protein interactions. *PLOS Comput. Boil.* **2018**, *14*, e1006616. [CrossRef]
17. Bellucci, M.; Agostini, F.; Masin, M.; Tartaglia, G.G. Predicting protein associations with long noncoding RNAs. *Nat. Methods* **2011**, *8*, 444–445. [CrossRef]
18. Lu, Q.; Ren, S.; Lu, M.; Zhang, Y.; Zhu, D.; Zhang, X.; Li, T. Computational prediction of associations between long non-coding RNAs and proteins. *BMC Genom.* **2013**, *14*, 651. [CrossRef]
19. Suresh, V.; Liu, L.; Adjeroh, D.; Zhou, X. RPI-Pred: Predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Res.* **2015**, *43*, 1370–1379. [CrossRef] [PubMed]
20. Hu, H.; Zhu, C.; Ai, H.; Zhang, L.; Zhao, J.; Zhao, Q.; Liu, H. LPI-ETSLP: lncRNA–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction. *Mol. BioSyst.* **2017**, *13*, 1781–1787. [CrossRef] [PubMed]

21. Zhao, Q.; Zhang, Y.; Hu, H.; Ren, G.; Zhang, W.; Liu, H. IRWNRLPI: Integrating Random Walk and Neighborhood Regularized Logistic Matrix Factorization for lncRNA-Protein Interaction Prediction. *Front. Genet.* **2018**, *9*, 239. [CrossRef]
22. Zhang, Y.; Wang, M.; Li, A.; Ge, M.; Peng, C. Predicting Long Noncoding RNA and Protein Interactions Using Heterogeneous Network Model. *BioMed Res. Int.* **2015**, *2015*. [CrossRef]
23. Ge, M.; Li, A.; Wang, M. A Bipartite Network-based Method for Prediction of Long Non-coding RNA-protein Interactions. *Genom. Proteom. Bioinform.* **2016**, *14*, 62–71. [CrossRef]
24. Liu, H.; Ren, G.; Hu, H.; Zhang, L.; Ai, H.; Zhang, W.; Zhao, Q. LPI-NRLMF: lncRNA-protein interaction prediction by neighborhood regularized logistic matrix factorization. *Oncotarget* **2017**, *8*, 103975–103984. [CrossRef] [PubMed]
25. Zhang, W.; Li, R.; Zeng, T.; Sun, Q.; Kumar, S.; Ye, J.; Ji, S. Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. *IEEE Trans. Big Data* **2016**. [CrossRef]
26. Zhang, L.; Lv, C.; Jin, Y.; Cheng, G.; Fu, Y.; Yuan, D.; Tao, Y.; Guo, Y.; Ni, X.; Shi, T. Deep Learning-Based Multi-Omics Data Integration Reveals Two Prognostic Subtypes in High-Risk Neuroblastoma. *Front. Genet.* **2018**, *9*, 477. [CrossRef]
27. Fuentes, A.F.; Yoon, S.; Lee, J.; Park, D.S. High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank. *Front. Plant Sci.* **2018**, *9*, 1162. [CrossRef]
28. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 346. [CrossRef]
29. Liu, G.; Bao, H.; Han, B. A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis. *Math. Probl. Eng.* **2018**, *2018*, 1–10. [CrossRef]
30. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
31. Alipanahi, B.; Delong, A.; Weirauch, M.T.; Frey, B.J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **2015**, *33*, 831–838. [CrossRef]
32. Zhou, J.; Troyanskaya, O.G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **2015**, *12*, 931–934. [CrossRef]
33. Wang, Y.; You, Z.-H.; Yang, S.; Li, X.; Jiang, T.-H.; Zhou, X. A High Efficient Biological Language Model for Predicting Protein-Protein Interactions. *Cells* **2019**, *8*, 122. [CrossRef]
34. Pan, X.; Shen, H.-B. Predicting RNA-protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics* **2018**, *34*, 3427–3436. [CrossRef]
35. Hosseini-Asl, E.; Zurada, J.M.; Nasraoui, O. Deep Learning of Part-Based Representation of Data Using Sparse Autoencoders With Nonnegativity Constraints. *IEEE Trans. Neural Networks Learn. Syst.* **2016**, *27*, 1–13. [CrossRef]
36. Halkias, X.; Paris, S.; Glotin, H. Sparse Penalty in Deep Belief Networks: Using the Mixed Norm Constraint. Available online: https://arxiv.org/abs/1301.3533 (accessed on 27 December 2018).
37. Vincent, P.; LaRochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
38. Van Der Laan, M.J.; Polley, E.C.; Hubbard, A.E. Super learner in prediction. U.C. Berkeley Division of Biostatistics Working Paper Series. Available online: https://biostats.bepress.com/ucbbiostat/paper222/ (accessed on 1 November 2018).
39. Muppirala, U.K.; Honavar, V.G.; Dobbs, D. Predicting RNA-Protein Interactions Using Only Sequence Information. *BMC Bioinform.* **2011**, *12*, 489. [CrossRef]
40. Wang, Y.; Xu, D.; Chen, R.; Chen, L.; Wang, Y.; Chen, X.; Liu, Z.-P.; Huang, Q.; Zhang, X.-S. De novo prediction of RNA–protein interactions from sequence information. *Mol. BioSyst.* **2013**, *9*, 133–142. [CrossRef]
41. Pan, X.; Fan, Y.-X.; Yan, J.; Shen, H.-B. IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction. *BMC Genom.* **2016**, *17*, 861. [CrossRef]
42. Yi, H.-C.; You, Z.-H.; Huang, D.-S.; Li, X.; Jiang, T.-H.; Li, L.-P. A Deep Learning Framework for Robust and Accurate Prediction of ncRNA-Protein Interactions Using Evolutionary Information. *Mol. Ther. Nucleic Acids* **2018**, *11*, 337–344. [CrossRef]
43. Cao, Z.; Pan, X.; Yang, Y.; Huang, Y.; Shen, H.-B.; Zhen, C. The lncLocator: a subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier. *Bioinformatics* **2018**, *34*, 2185–2194. [CrossRef]

44. Hu, H.; Zhang, L.; Ai, H.; Zhang, H.; Fan, Y.; Zhao, Q.; Liu, H. HLPI-Ensemble: Prediction of human lncRNA-protein interactions based on ensemble strategy. *RNA Boil.* **2018**, *15*, 1–10. [CrossRef]
45. Xiao, Y.; Wu, J.; Lin, Z.; Zhao, X. A deep learning-based multi-model ensemble method for cancer prediction. *Comput. Methods Programs Biomed.* **2018**, *153*, 1–9. [CrossRef]
46. Liu, Z.-P.; Miao, H. Prediction of protein-RNA interactions using sequence and structure descriptors. *Neurocomputing* **2016**, *206*, 28–34. [CrossRef]
47. Liu, B. BioSeq-Analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings Bioinform.* **2017**, 165. [CrossRef] [PubMed]
48. Le, Q.V. Building high-level features using large scale unsupervised learning. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8595–8598.
49. Sankaran, A.; Vatsa, M.; Singh, R.; Majumdar, A. Group sparse autoencoder. *Image Vision Comput.* **2017**, *60*, 64–74. [CrossRef]
50. Zhang, X.; Liu, S. RBPPred: Predicting RNA-binding proteins from sequence using SVM. *Bioinformatics* **2017**, *33*, 854–862. [CrossRef] [PubMed]
51. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc: Red Hook, NY, USA, 2017; pp. 3146–3154.
52. Zhan, Z.H.; You, Z.H.; Li, L.P.; Zhou, Y.; Yi, H.C. Accurate Prediction of ncRNA-Protein Interactions From the Integration of Sequence and Evolutionary Information. *Front Genet.* **2018**, *9*, 458. [CrossRef] [PubMed]
53. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
54. Wang, H.; Liu, C.; Deng, L. Enhanced Prediction of Hot Spots at Protein-Protein Interfaces Using Extreme Gradient Boosting. *Sci. Rep.* **2018**, *8*, 14285. [CrossRef]
55. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Available online: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (accessed on 30 October 2018).
56. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Networks* **1999**, *12*, 145–151. [CrossRef]
57. Arendsee, Z.W.; Li, L.; Wurtele, E.S. Coming of age: orphan genes in plants. *Trends Plant. Sci.* **2014**, *19*, 698–708. [CrossRef]

Sample Availability: The code and datasets for implementing the proposed method are available at https://github.com/Mjwl/PLRPIM.