SCIENTIFIC REPORTS

OPEN

Deep-RBPPred: Predicting RNA binding proteins in the proteome scale based on deep learning

Received: 17 May 2018
Accepted: 28 September 2018
Published online: 15 October 2018

Jinfang Zheng, Xiaoli Zhang, Xunyi Zhao, Xiaoxue Tong, Xu Hong, Juan Xie & Shiyong Liu

RNA binding protein (RBP) plays an important role in cellular processes. Identifying RBPs by computation and experiment are both essential. Recently, an RBP predictor, RBPPred, is proposed in our group to predict RBPs. However, RBPPred is too slow for that it needs to generate PSSM matrix as its feature. Herein, based on the protein feature of RBPPred and Convolutional Neural Network (CNN), we develop a deep learning model called Deep-RBPPred. With the balance and imbalance training set, we obtain Deep-RBPPred-balance and Deep-RBPPred-imbalance models. Deep-RBPPred has three advantages comparing to previous methods. (1) Deep-RBPPred only needs few physicochemical properties based on protein sequences. (2) Deep-RBPPred runs much faster. (3) Deep-RBPPred has a good generalization ability. In the meantime, Deep-RBPPred is still as good as the state-of-the-art method. Testing in A. thaliana, S. cerevisiae and H. sapiens proteomes, MCC values are 0.82 (0.82), 0.65 (0.69) and 0.85 (0.80) for balance model (imbalance model) when the score cutoff is set to 0.5, respectively. In the same testing dataset, different machine learning algorithms (CNN and SVM) are also compared. The results show that CNN-based model can identify more RBPs than SVM-based. In comparing the balance and imbalance model, both CNN-base and SVM-based tend to favor the majority class in the imbalance set. Deep-RBPPred forecasts 280 (balance model) and 265 (imbalance model) of 299 new RBP. The sensitivity of balance model is about 7% higher than the state-of-the-art method. We also apply deep-RBPPred to 30 eukaryotes and 109 bacteria proteomes downloaded from Uniprot to estimate all possible RBPs. The estimating result shows that rates of RBPs in eukaryote proteomes are much higher than bacteria proteomes.

RNA binding proteins (RBPs) play important functions in many cellular processes, such as post-transcriptional gene regulation, RNA subcellular localization and alternative splicing. With significant function in biology, many high-throughput experimental techniques have been developed to identify new RBPs in human, mouse, S. cerevisiae and C. elegans[1–10]. After RBPs have been identified, CLIP-related experimental technologies[11–14] are applied to reveal the binding sites in RNAs. Also, many computational methods have been proposed to predict interaction of protein with RNA[15–18] and RBPs[19–25]. RBP predictors can predict the RBPs, and then CLIP-related techniques can further reveal RNAs interacting with these RBPs. However, previous computational methods only considered only part features or known RNA binding domain (RBD) which plays a significant role in RBPs prediction. So, we proposed RBPPred integrating as much as features to address this problem[22]. Benchmarking on datasets shows that RBPPred is better than other approaches. But RBPPred runs slowly because it requires to run blast against a huge protein NR database to generate PSSM matrix. However, the prediction speed is important because a large number of RBPs are still unknown in many species. To overcome this shortcoming, we present Deep-RBPPred which is based on deep learning.

In recently years, deep learning technology has been used in many aspects in bioinformatics and proved as a power tool[26–32]. For predicting protein binding sites in RNA sequence, DeepBind[32] is the first CNN-based model to predict the binding affinity. Deep-rbp[29] and iDeep[30,31] are two deep learning methods which both take RNA structure into consideration. These methods outperform the conventional approaches in term of prediction accuracy. However, deep learning algorithm is still not applied to RBPs prediction. In Deep-RBPPred, we apply a deep convolutional neural network instead of SVM. Since CNN-based method requires to input a fixed length feature vector, two solutions are handled to meet this requirement. The first solution is to pad all the sequences to fixed length sequences, and then one-hot encoding is used to encode the sequences. The second solution is to design

School of Physics, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China. Correspondence and requests for materials should be addressed to S.L. (email: liushiyong@gmail.com)

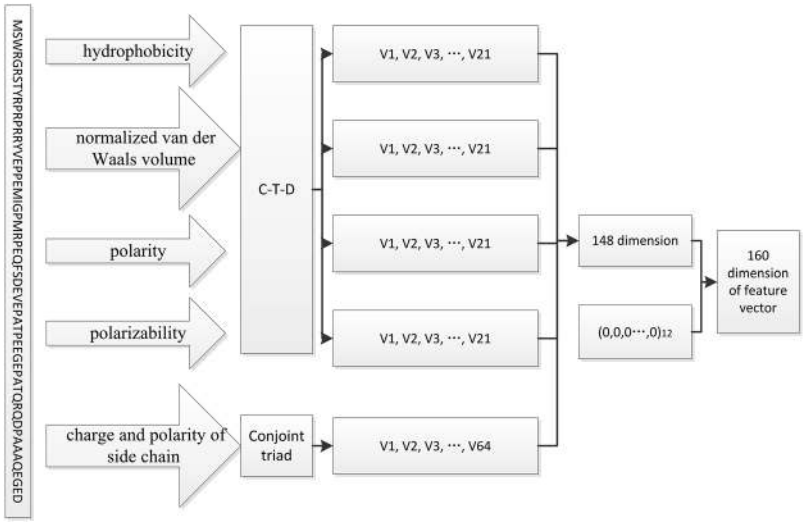

Figure 1. Process of encoding a protein sequence into the 160 dimension feature vector. For the properties of hydrophobicity, polarity, normalized van der Waals volume, polarizability, the global protein sequence descriptors (C-T-D) was employed to encode each feature vector with 21 dimension (v1, v2, v3, v4, …, v21). According to charge and polarity of side chain, the protein sequence was encoded to a vector of 64 dimension (v1, v2, v3, v4, …, v64) through the conjoint triad encoding method. This process is a part of RBPPred encoding process[22].

the features by hand. It is not appropriate to predict RBPs with the padding solution because the length of RBPs varies over a wide range (50–10 K, see methods). Based on this consideration, we employ the hand-designed features which are proved effective to represent RBPs in RBPPred. Unlike RBPPred, we only employ physicochemical features including hydrophobicity, polarity, normalized van der Waals volume, polarizability, side chain's charge and polarity. These features are used to train the weights of 11 layers convolutional neural network with Tensorflow[33]. Deep-RBPPred presents comparable results to RBPPred but is significantly more efficient in the testing. And it also only needs few physicochemical features.

Methods

Training sets. In order to train our deep learning model, we employed the training set used in the RBPPred. The details of generating the training set have been described in RBPPred[22]. Here we just simply describe the generating process. The positive samples are collected from the Uniprot database[34], which is retrieval with GO term'RNA binding' to search this protein database because the Uniprot database includes RBPs with x-ray crystal structures and RBPs identified by high-through experiment. For the negative samples, we took the approach from SPOT-stru[35]. The negative sequences are collected from PDB, by using PISCCES[36] with sequence identity cutoff 25%, sequence length between 50 and 10,000 and resolution of X-ray better than 3.0 Å. The collected sequences are then mixed together so that the redundant proteins are removed with sequence identity of 25% by psi-cd-hit in the CD-HIT package[37]. Finally, the training set includes 2780 RBPs and 7093 non-RBPs.

The training set consisting of different amount of RBPs and non-RBPs is known as an imbalance training set. The classification algorithm tends to favor the majority class when it is trained in the imbalance dataset. So, we randomly select 2780 non-RBPs to generate the balance dataset together with 2780 RBPs.

Testing set. For testing our deep learning model, we used the testing set from RBPPred[22]. However, only the identical sequences between the training and the testing set are removed. This may lead to a bias result caused by the redundance between training and testing set. So, we remove the homology sequences by CD-HIT. The testing and training set are mixed together to be clustered by CD-HIT with sequence identity cutoff 30%. Then all the sequences are discarded from testing set if they are in the same cluster with the training sequences. This can ensure the testing sequences are independent with training set. For one cluster, we only select the sequence provided by CD-HIT to ensure a non-redundant testing set. We finally collected 488 sequences including 239 negative samples and 249 positive samples, which are composed of 72 RBPs and 13 non-RBPs for A. thaliana, 129 RBPs and 164 non-RBPs for H. sapiens, 48 RBPs and 62 non-RBPs for S. cerevisiae. Comparing to the previous testing dataset including 2546 sequences, 2058 sequences are discarded for the redundance.

Protein features and encoding. The protein is encoded to a feature vector by the approach described in RBPPred[37]. But the evolutionary information and predicted secondary structure are discarded due to the computational time. The solvent accessibility is also discarded. At last, a 148-dimensional vector is encoded to represent each protein sequence including the properties of hydrophobicity, normalized van der Waals volume, polarity and polarizability, charge and polarity of side chain. We expand the dimension of feature vector to 160 with the expanded feature values assigned to 0 due to the CNN network architecture. Finally, we collect a total 160-dimensional feature vector to represent a protein sequence, as shown in Fig. 1. The detail of encoding the

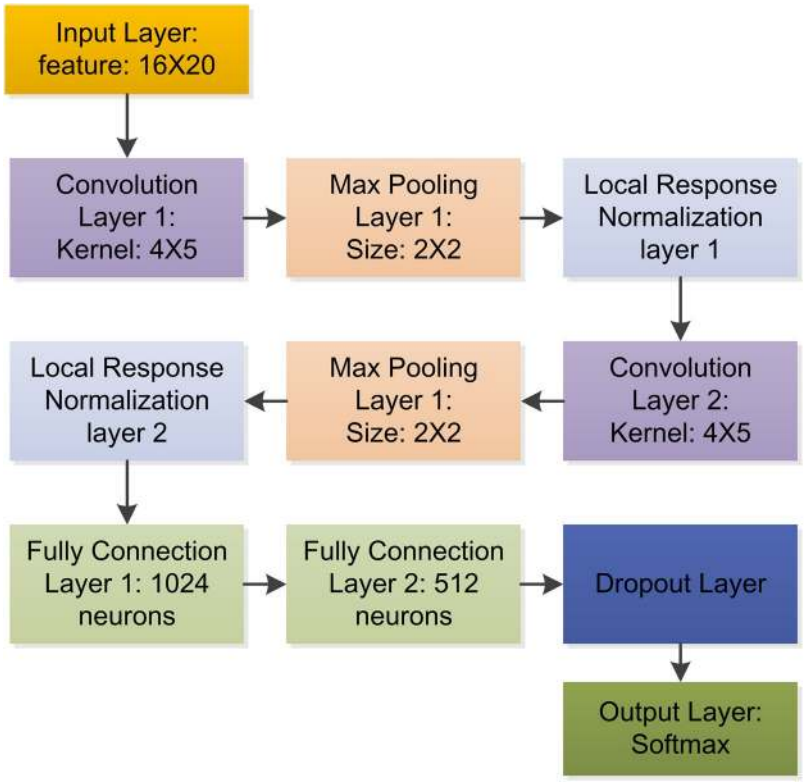

Figure 2. Network architecture of Deep-RBPPred. Deep-RBPPred is a CNN network including 11 layers. Convolution Layer and Max Pooling Layer are designed to automatically process the feature. Local Response Normalization layer and Dropout Layer are designed to avoid over-fitting. Softmax layer is used to classify the protein with probability score. The batch size is assigned to 200. The learning rate is set to 0.0001.

physicochemical properties is described in RBPPred[22]. The program encoding the feature vector is extracted from the RBPPred software package.

Performance evaluation. The performance is evaluated by sensitivity (SN), specificity (SP), accuracy (ACC) and Matthews Correlation Coefficient (MCC) which are defined as following:

$$\text{Sensitivity (SN)} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Specificity (SP)} = \text{TN}/(\text{TN} + \text{FP})$$

$$\text{Accuracy (ACC)} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FN} + \text{FP})$$

$$\text{Matthews Correlation Coefficient (MCC)} = (\text{TP}^{*}\text{TN} - \text{FP}^{*}\text{FN})/\text{sqrt}((\text{TP} + \text{FN})^{*}(\text{TP} + \text{FP})^{*}(\text{TN} + \text{FP})^{*}(\text{TN} + \text{FN}))$$

where, TP is true positive, and FN is false negative. TN refers to true negative and FP refers to false positive. The AUC is also applied to measure the performance.

Network architecture of Deep-RBPPred. Deep-RBPPred is a Convolutional Neural Network (CNN)[38] with tensorflow. In Fig. 2, it shows the network architecture of Deep-RBPPred. The protein feature vector is reshaped to a tensor with shape 8×20 in order to apply the 2D-convolution function. So the input layer is a size of 8×20 feature tensor representing a protein. The following layer is a convolution layer with a kernel size of 2×5. In this layer, 32 convolution kernels are set to filter the input features. The third layer is a max pooling layer with a size of 2×2. The feature size will be reduced to 4×10 after the layer. And the next layer is a local response normalized layer. This layer is set to increase the generalization ability. The following three layers are convolution layer, max pooling layer and local response normalized layer, respectively. Then the feature tensor is flatted to a 640-dimensional vector. The following two layers are fully connected layers with 512 and 256 neurons, respectively. The 10th layer is a dropout layer[39] which randomly discards some neurons in the training phase. The dropout probability is set to 0.5. The final layer is the Softmax layer which is used to classify RBPs or not. The output of this model is a probability score which describes the probability of an RBP. All the activation functions in neurons are ReLU[40]. All the weights in neurons are added a L2 regularization operation. The L2 regularization losses are

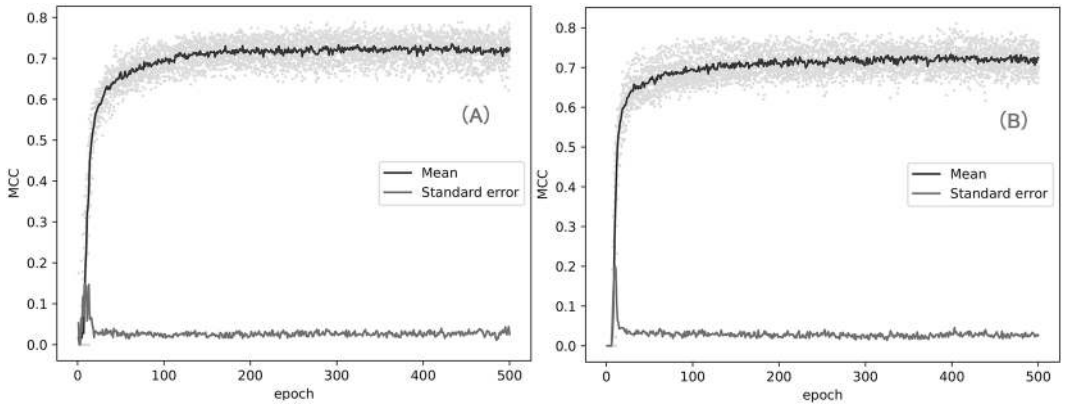

Figure 3. Training process of 10-fold cross-validation on balance set (**A**) and imbalance set (**B**). In 10-fold cross-validation, we calculate the mean MCC of each epoch and the standard error of MCC. For balance set, the highest average MCC is 0.74 and the highest standard error is 0.15. For imbalance set, the highest average MCC is 0.73 and the highest standard error is 0.20.

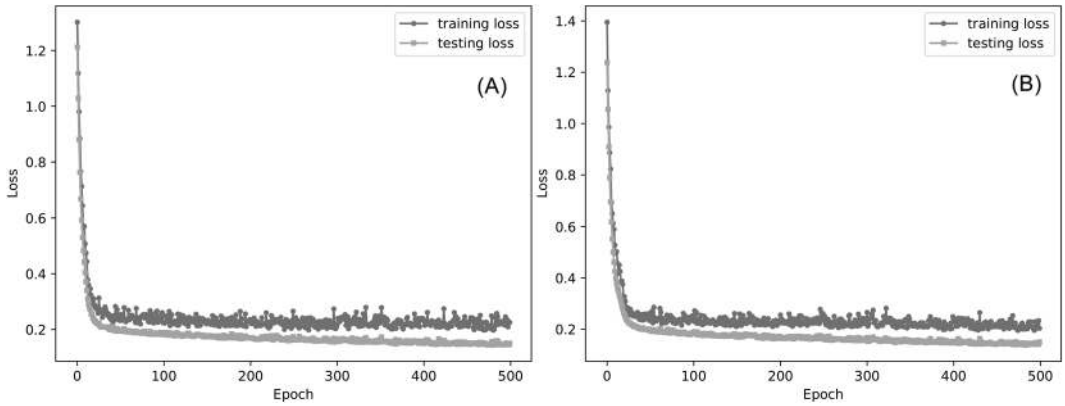

Figure 4. The training process in imbalance (**A**) and balance training set (**B**). The loss is defined as the sum of L2 regularization loss and the cross entropy (see text).

added to the final loss function. Adam optimizer is employed to minimize final loss consisting of cross-entropy between the label and probability score and L2 regularization loss of neurons. In this architecture, the number of trainable variable is 480,930. In training process, the learning rate is set to 0.0001.

Result

10-fold cross-validation on the training set. To avoid the overfitting and estimate an appropriate epoch of our models in the whole training sets, we perform the 10-fold cross-validation on the balance and imbalance training set. As shown in Fig. 3, the balance model converges between 100 and 200 epochs. The imbalance model converges between 300 and 400 epochs. Comparing to the balance model, the imbalance converges at a later epoch. This may be caused by the more sequences in the imbalance training set. Figure 3 also shows that the balance model achieves a slighter higher MCC than the imbalance model. The result of 10-fold cross-validation indicates that 500 epochs can be used in the training process and avoid the overfitting caused by a higher epoch. The result of 10-fold cross-validation also indicates that the parameters of network (batch size, the number of neuron, learning rate) can work in this model.

Training process and model selection. In order to achieve models constructed on the whole training sets, the CNN network is trained in the balance and imbalance training set with the epoch determined in the 10-fold cross-validation. The training and testing loss are used to evaluate the models in each epoch. As shown in Fig. 4, the training and testing loss decrease with the epoch. This training processes are similar to the 10-fold cross-validation (Fig. S1). The testing loss decreases rapidly at the early epoch and then the value remains the same after the convergence point. In theory, all models near the convergence point can be used as the final model. In order to get the best prediction performance, the balance model of 390th epoch and imbalance model of 242th epoch are selected as final models. The training loss of final balance/imbalance model is 0.15/0.17, which is almost equal to loss of the 10-folds validation process (Fig. S1). The testing loss of final balance/imbalance is 0.23/0.24. This indicates that our models is not overfitting.

Model	SVM-imbalance			SVM-balance		
Dataset	S	H	A	S	H	A
ACC	0.75	0.80	0.71	0.76	0.74	0.85
SN	0.56	0.64	0.65	0.83	0.81	0.83
SP	0.90	0.93	1.0	0.71	0.70	0.92
AUC	0.86	0.88	0.93	0.85	0.85	0.92
MCC	0.50	0.60	0.47	0.54	0.50	0.60

Table 1. Performance on the testing set for the SVM-based model. *H, S, and A stand for H. sapiens, S. cerevisiae, and A. thaliana species respectively.

Model	Deep-RBPPred-balance			Deep-RBPPred-imbalance			RBPPred		
Dataset	H	S	A	H	S	A	H	S	A
ACC	0.91	0.81	0.95	0.91	0.85	0.94	0.91	0.88	0.90
SN	0.96	0.94	0.94	0.89	0.83	0.94	0.85	0.85	0.88
SP	0.87	0.71	1.0	0.93	0.85	0.92	0.96	0.90	1.0
AUC	0.97	0.90	0.99	0.96	0.91	0.98	0.98	0.95	0.98
MCC	0.83	0.65	0.85	0.82	0.69	0.80	0.81	0.76	0.72

Table 2. Performance comparison on the testing set. *H, S, and A stand for H. sapiens, S. cerevisiae, and A. thaliana species respectively.

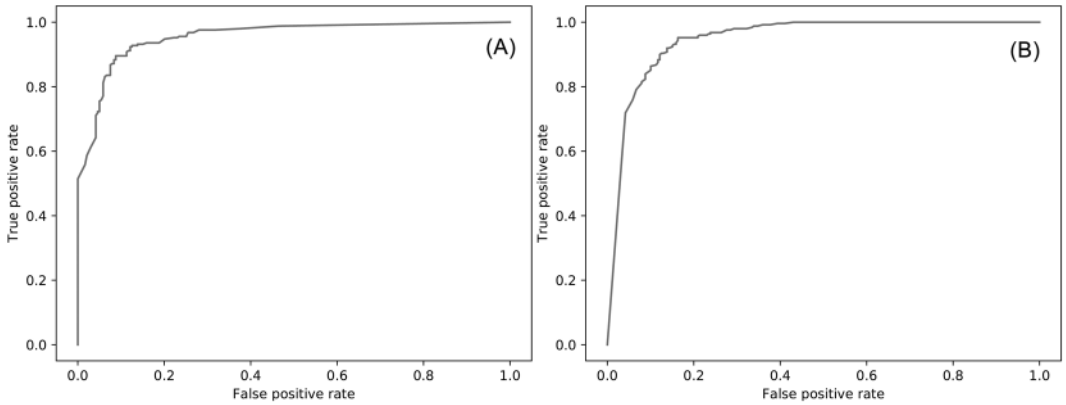

Figure 5. ROC for Deep-RBPPred-imbalance (**A**) and Deep-RBPPred-balance (**B**). The AUC for balance/imbalance model is 0.95/0.95.

Performance in independent testing and comparison to other models. In this section, we comprehensively evaluate our deep learning models in a non-redundant testing dataset. Firstly, we compare the power of two of the most popular machine learning algorithms, SVM and CNN, in RBPs prediction. Secondly, the balance and imbalance models are compared to reveal the affections of these two models in predicting RBPs. Thirdly, three RBPs prediction approaches, SONAR[25], RNApred[23] and RBPPred[27] are compared with our deep learning models in a non-redundant testing dataset. All results are shown in Tables 1–3 and Fig. 5. The ROC curves are plotted in Figs S2 and S3.

In order to make a comparison of the power of machine learning algorithm in predicting RBPs, we also train a balance model and an imbalance model with SVM. The SVM-based models are constructed on the training sets with the libsvm-3.22[41] and tested in the testing dataset. The results are shown in Table 1. As shown, the SVM-imbalance model achieves MCC values of 0.50, 0.60 and 0.47 for S. cerevisiae, H. sapiens and A. thaliana. The SVM-balance model achieves MCC values of 0.54, 0.50 and 0.60 for S. cerevisiae, H. sapiens and A. thaliana. The result indicates there is no significant difference for the imbalance and balance model. In Table 2, Deep-RBPPred-balance achieves MCC values of 0.82 for H. sapiens, 0.69 for S. cerevisiae, 0.80 for A. thaliana, which are both much higher than the balance and imbalance model of SVM.

For non-RBPs in the testing set, the SVM-balance model obtains an average SP of 0.78 ((0.71 + 0.70 + 0.92)/3), which is much better than average SP of 0.62 ((0.56 + 0.64 + 0.65)/3) for the imbalance model. For predicting RBPs in the testing dataset, the SVM-imbalance model achieves an average of SN 0.94 ((0.90 + 0.93 + 1.0)/3), which is much higher than average SN of 0.82 ((0.83 + 0.81 + 0.83)/3) for the balance model. Indeed, the SVM models trained on the balance or imbalance model have an effect on the non-RBPs prediction. That is, the balance model has a better/worse predicting ability in RBPs/non-RBPs than the imbalance model. This

Method	RNApred			SONAR
Dataset	S	H	A	H
ACC	0.65	0.68	0.86	0.88
SN	0.90	0.88	0.93	0.92
SP	0.45	0.52	0.46	0.85
AUC	0.82	0.80	0.83	0.84
MCC	0.38	0.41	0.42	0.77

Table 3. Performance of RNApred in testing dataset. *H, S, and A stand for H. sapiens, S. cerevisiae, and A. thaliana species respectively. SONAR is developed for the human, and gene name is used as input (not protein sequence). The gene name may be the same between different species, so we only test the performance in human proteome.

Proteome	UP000000559	UP000005640	UP000000589	UP000006548
Times (CPU)	4 s	40 s	33 s	30 s
Times (GPU)	3 s	14 s	13 s	12 s

Table 4. Computational time of Deep-RBPPred running in Centos with Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10 GHz and GeForce GTX 1080Ti. *UP000000559, UP000005640, UP000000589 and UP000006548 include 1000, 20231, 16946 and 15524 protein sequences. Reviewed proteomes are downloaded from the Uniprot.

effect is not significant in CNN when comparing with SVM. As shown in Table 2, Deep-RBPPred-balance achieves an average SN of 0.95 ((0.96 + 0.94 + 0.94)/3). For non-RBPs, Deep-RBPPred-balance achieves an average SP of 0.86 ((0.87 + 0.71 + 1.0)/3), which is lower than average SP of 0.90 ((0.93 + 0.85 + 0.92)/3) of Deep-RBPPred-imbalance. Figure 5 shows the ROC of Deep-RBPPred.

Deep-RBPPred is also tested and compared to other models on the testing set. Total three predictors are compared. The first approach is RBPPred, which is developed by our group previously. The second approach is RNApred which employees the amino acid composition or PSSM to predict RBPs. The third method is SONAR which integrates protein-protein interaction network and other features to predict the RBPs. The result of SPOT-Seq-RNA is not shown here because it has been compared with RBPPred[22].

As shown in Table 2, Deep-RBPPRed-balance achieves MCC values of 0.83, 0.65 and 0.85 for H. sapiens, S. cerevisiae, A. thaliana, respectively. The performance of the imbalance model of Deep-RBPPred is almost as good as the balance model. RBPPred achieves MCC values of 0.81, 0.76 and 0.72 for H. sapiens, S. cerevisiae, A. thaliana respectively. We can find that RBPPred and Deep-RBPPred have different performances in S. cerevisiae and A. thaliana proteomes. The average MCC of Deep-RBPPred-balance ((0.83 + 0.65 + 0.85)/3) has a value almost as high as RBPPred ((0.81 + 0.76 + 0.72)/3). As shown in Table 3, Deep-RBPPred achieves a much higher MCC than RNApred ((0.38 + 0.41 + 0.42)/3). Deep-RBPPred also performs better than SONAR in the human proteome.

Capacity of prediction new RBPs. To test the predicting ability of Deep-RBPPred on new RBPs, we collect 299 new RBPs created between 2015-05-24 (consistent with RBPPred) and 2017-09-27 from Uniprot. In this section, only the RBPPred is compared because that the RBPPred have been proved to have better predicting ability than other methods[22]. 280 and 265 of 299 new RBPs are correctly predicted by Deep-RBPPred-balance and Deep-RBPPred-imbalance, but only 260 RBPs are predicted by RBPPred[22]. Deep-RBPPred performs better than RBPPred. One protein (Uniprotid: P0DOC6) can't be calculated by RBPPred for that no protein sequences can be found by Blast. We also collect the 130 experimently determined human RBPs published in Wen and the co-workers' work[42]. RBPPred correctly predicts 24 of 130 RBPs, while Deep-RBPPred-imbalance and Deep-RBPPred-balance correctly predict 63 and 92 RBPs respectively. These results indicate that Deep-RBPPRed has better predicting ability than RBPPred.

Computational time. Running time is an important metric to measure a model. We list the computational time of Deep-RBPPred in Table 4. The table shows that Deep-RBPPred is a very fast RBP predictor. Here we do not list the computational time of RBPPred because it costs much more computational time. Comparing with RBPPred, Deep-RBPPred predicts RBPs without using blast to generate PSSM matrix which is a time-consuming step. Take the advantage of computational time, Deep-RBPPred can be used to estimate RBPs in proteome scale quickly.

RBPs estimation in the 139 reviewed proteomes. Deep-RBPPred is applied to estimate the RBPs in 139 reviewed proteomes for 109 bacteria and 30 eukaryote species. There are two problems in the Uniprot proteome dataset. Firstly, reviewed and un-reviewed sequences are both included in the Uniprot. The un-reviewed sequence may not be a real protein. So, the reviewed proteomes are used in the prediction. The second problem is that almost all reviewed proteomes are incomplete. For example, truepera radiovictrix (proteome id: UP000000379) only contained one reviewed sequence. These two problems can result in a bias prediction. In

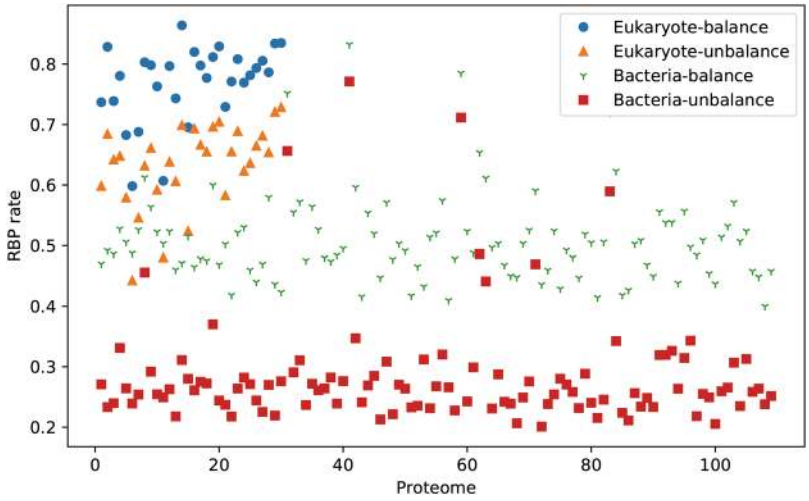

Figure 6. The RBPs rate estimation of Deep-RBPPred in the reviewed eukaryota and bacteria proteomes from uniprot. Total 109 bacteria and 30 eukaryote proteome are kept due to the limitation of sequence amount in the proteome (1/10 of yeast for eukaryote and 1/10 of *E. coli* for bacteria). The labels 'Eukaryote-balance' stands for that the eukaryotes proteome is predicted by the balance model.

order to select some appropriate proteomes, we filter the proteomes with the number of sequences. For eukaryote species, the amount is set to 1/10 of S. cerevisiae. For bacteria, the number is set to 1/10 of *E. coli*. Finally, we filter out 109 bacteria and 30 eukaryote proteomes from all the bacteria and eukaryote proteomes.

The results of prediction are shown in Fig. 6. The balance model predicts more RBPs than the imbalance model in bacteria and eukaryote. For the predicting results with the balance model and imbalance model, we found an interesting phenomenon that the rate of RBPs in eukaryotes proteome is higher than bacteria. This result implies RBPs may function in more complex cellular processes in eukaryotes. For the human proteome, we estimate 14,744 RBPs with the imbalance model.

Discussion

In this study, we develop two RBPs predicting models (the balance and imbalance model) based on CNN which only need hydrophobicity, normalized van der waals volume, polarity and polarizability, charge and polarity of side chain of protein sequence. Comparing with SVM models, we show that the CNN-based model performs better than SVM-based model. In comparing the balance and imbalance model, both the CNN-based and SVM-based classification show a preference for the major class. The result from the testing dataset shows that our deep learning models perform as good as RBPPred which is the best model so far. More importantly, Deep-RBPPred needs fewer features than RBPPred. Deep-RBPPred was then applied to estimate RBPs in 139 reviewed proteomes from the Uniprot dataset. The result shows that the RBPs rate in the bacteria is smaller than the eukaryote proteome. Deep-RBPPred-imbalance predicts 14,744 RBPs for the whole reviewed human proteomes. This number is almost 10-fold more than the number of the RBPs identified by high through experiments. This may be caused by these high throughput experimental limitations. For example, "Interactome Capture" only identifies the RBPs which bind to mRNA[2]. It may lose the RBPs binding to the non-coding RNA.

In general, deep learning methods are applied in a large-scale data. A classic application of CNN-based methods is to classify the image. In this application, data augmentation approaches are used to enlarge the number of samples. And a large dataset may reduce the risk of deep learning model in overfitting. In Deep-RBPPred, we remove the redundant sequences as other researches have done. The process can be regarded as the opposite of data augmentation process in image recognition. In addition, L2 regularization and dropout layer[43] are added to avoid overfitting in the architecture of our deep learning. The process of 10-cross validation (Fig. 3) shows the MCC is almost no longer increases after 100th epoch. The process of training (Fig. 4) also shows the training/testing loss does not change too much round 0.2/0.14. These phenomena imply Deep-RBPPred is not overfitting. The real number of RBPs is still unknown and new RBPs are discovered as time goes by. Our prediction may benefit the RBP community.

Data Availability

Deep-RBPPred is written in the python, availability as an open source tool at http://www.rnabinding.com/Deep_RBPPred/Deep-RBPPred.html.

References

1. Baltz, A. G. *et al*. The mRNA-bound proteome and its global occupancy profile on protein-coding transcripts. *Mol Cell* **46**, 674–690, https://doi.org/10.1016/j.molcel.2012.05.021 (2012).
2. Castello, A. *et al*. Insights into RNA biology from an atlas of mammalian mRNA-binding proteins. *Cell* **149**, 1393–1406, https://doi.org/10.1016/j.cell.2012.04.031 (2012).

3. Kwon, S. C. *et al*. The RNA-binding protein repertoire of embryonic stem cells. *Nat Struct Mol Biol* **20**, 1122–1130, https://doi.org/10.1038/nsmb.2638 (2013).
4. Mitchell, S. F., Jain, S., She, M. & Parker, R. Global analysis of yeast mRNPs. *Nat Struct Mol Biol* **20**, 127–133, https://doi.org/10.1038/nsmb.2468 (2013).
5. Wessels, H. H. *et al*. The mRNA-bound proteome of the early fly embryo. *Genome Res* **26**, 1000–1009, https://doi.org/10.1101/gr.200386.115 (2016).
6. Bunnik, E. M. *et al*. The mRNA-bound proteome of the human malaria parasite Plasmodium falciparum. *Genome Biol* **17**, 147, https://doi.org/10.1186/s13059-016-1014-0 (2016).
7. Beckmann, B. M. *et al*. The RNA-binding proteomes from yeast to man harbour conserved enigmRBPs. *Nat Commun* **6**, 10127, https://doi.org/10.1038/ncomms10127 (2015).
8. Matia-Gonzalez, A. M., Laing, E. E. & Gerber, A. P. Conserved mRNA-binding proteomes in eukaryotic organisms. *Nat Struct Mol Biol* **22**, 1027–1033, https://doi.org/10.1038/nsmb.3128 (2015).
9. Liao, Y. *et al*. The Cardiomyocyte RNA-Binding Proteome: Links to Intermediary Metabolism and Heart Disease. *Cell Rep* **16**, 1456–1469, https://doi.org/10.1016/j.celrep.2016.06.084 (2016).
10. Liepelt, A. *et al*. Identification of RNA-binding Proteins in Macrophages by Interactome Capture. *Mol Cell Proteomics* **15**, 2699–2714, https://doi.org/10.1074/mcp.M115.056564 (2016).
11. Licatalosi, D. D. *et al*. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature* **456**, 464–469, https://doi.org/10.1038/nature07488 (2008).
12. Konig, J. *et al*. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nat Struct Mol Biol* **17**, 909–915, https://doi.org/10.1038/nsmb.1838 (2010).
13. Hafner, M. *et al*. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell* **141**, 129–141, https://doi.org/10.1016/j.cell.2010.03.009 (2010).
14. Van Nostrand, E. L. *et al*. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nat Methods* **13**, 508–514, https://doi.org/10.1038/nmeth.3810 (2016).
15. Bellucci, M., Agostini, F., Masin, M. & Tartaglia, G. G. Predicting protein associations with long noncoding RNAs. *Nat Methods* **8**, 444–445, https://doi.org/10.1038/nmeth.1611 (2011).
16. Muppirala, U. K., Honavar, V. G. & Dobbs, D. Predicting RNA-Protein Interactions Using Only Sequence Information. *Bmc Bioinformatics* **12**, https://doi.org/10.1186/1471-2105-12-489 (2011).
17. Suresh, V., Liu, L., Adjeroh, D. & Zhou, X. B. RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Res* **43**, 1370–1379, https://doi.org/10.1093/nar/gkv020 (2015).
18. Lu, Q. S. *et al*. Computational prediction of associations between long non-coding RNAs and proteins. *Bmc Genomics* **14**, https://doi.org/10.1186/1471-2164-14-651 (2013).
19. Zhao, H., Yang, Y. & Zhou, Y. Highly accurate and high-resolution function prediction of RNA binding proteins by fold recognition and binding affinity prediction. *RNA Biol* **8**, 988–996, https://doi.org/10.4161/rna.8.6.17813 (2011).
20. Yang, Y., Zhan, J., Zhao, H. & Zhou, Y. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. *Proteins* **80**, 2080–2088, https://doi.org/10.1002/prot.24100 (2012).
21. Paz, I., Kligun, E., Bengad, B. & Mandel-Gutfreund, Y. BindUP: a web server for non-homology-based prediction of DNA and RNA binding proteins. *Nucleic Acids Res* **44**, W568–574, https://doi.org/10.1093/nar/gkw454 (2016).
22. Zhang, X. & Liu, S. RBPPred: predicting RNA-binding proteins from sequence using SVM. *Bioinformatics* **33**, 854–862, https://doi.org/10.1093/bioinformatics/btw730 (2017).
23. Kumar, M., Gromiha, M. M. & Raghava, G. P. SVM based prediction of RNA-binding proteins using binding residues and evolutionary information. *J Mol Recognit* **24**, 303–313, https://doi.org/10.1002/jmr.1061 (2011).
24. Sharan, M., Forstner, K. U., Eulalio, A. & Vogel, J. APRICOT: an integrated computational pipeline for the sequence-based identification and characterization of RNA-binding proteins. *Nucleic Acids Res* **45**, e96, https://doi.org/10.1093/nar/gkx137 (2017).
25. Brannan, K. W. *et al*. SONAR Discovers RNA-Binding Proteins from Analysis of Large-Scale Protein-Protein Interactomes. *Mol Cell* **64**, 282–293, https://doi.org/10.1016/j.molcel.2016.09.003 (2016).
26. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res* **26**, 990–999, https://doi.org/10.1101/gr.200535.115 (2016).
27. Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* **32**, i121–i127, https://doi.org/10.1093/bioinformatics/btw255 (2016).
28. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods* **12**, 931–934, https://doi.org/10.1038/nmeth.3547 (2015).
29. Zhang, S. *et al*. A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res* **44**, e32, https://doi.org/10.1093/nar/gkv1025 (2016).
30. Pan, X. & Shen, H. B. Predicting RNA-protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/bty364 (2018).
31. Pan, X. & Shen, H. B. RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *BMC Bioinformatics* **18**, 136, https://doi.org/10.1186/s12859-017-1561-8 (2017).
32. 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* **33**, 831–838, https://doi.org/10.1038/nbt.3300 (2015).
33. Abadi, M. *et al*. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv 1603.04467* (2016).
34. UniProt, C. UniProt: a hub for protein information. *Nucleic Acids Res* **43**, D204–212, https://doi.org/10.1093/nar/gku989 (2015).
35. Zhao, H., Yang, Y. & Zhou, Y. Structure-based prediction of RNA-binding domains and RNA-binding sites and application to structural genomics targets. *Nucleic Acids Res* **39**, 3017–3025, https://doi.org/10.1093/nar/gkq1266 (2011).
36. Wang, G. & Dunbrack, R. L. Jr. PISCES: a protein sequence culling server. *Bioinformatics* **19**, 1589–1591 (2003).
37. Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658–1659, https://doi.org/10.1093/bioinformatics/btl158 (2006).
38. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).
39. Krizhevsky, A., Sutskever, I. & Hinton, G. E. In *Advances in neural information processing systems*. 1097–1105.
40. Glorot, X., Bordes, A. & Bengio, Y. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* 315–323.
41. Chang, C.-C. & Lin, C.-J. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* **2**, 27 (2011).
42. Qin, W. *et al*. Quantitative time-resolved chemoproteomics reveals that stable O-GlcNAc regulates box C/D snoRNP biogenesis. *Proc Natl Acad Sci USA* **114**, E6749–E6758, https://doi.org/10.1073/pnas.1702688114 (2017).
43. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).

Acknowledgements

We thank the National Supercomputer Center in Guangzhou for support of computing resources. This work has been supported by the Fundamental Research Funds for the Central Universities [2016YXMS017] and the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) under Grant No. U1501501.

Author Contributions

S.L. designed the project. J.Z. and S.L. wrote the main manuscript text. J.Z. did most computation and coding and data analysis and prepared all figures. X.Z. provided the dataset. X.Z. and X.Z., X.H. and J.X. did part of computation and data analysis. All authors reviewed and approved the final version of the manuscript.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-018-33654-x.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.