DeepPlnc: Discovering plant lncRNAs through multimodal deep learning on sequential data

Ritu[1,2,†], Sagar Gupta[1,†], Nitesh Kumar Sharma[1], Ravi Shankar[*,1,2]

[1] Studio of Computational Biology & Bioinformatics,

The Himalayan Centre for High-throughput Computational Biology

(HiCHiCoB, A BIC supported by DBT, Govt. of India)

CSIR-Institute of Himalayan Bioresource Technology (CSIR-IHBT),

Palampur (HP), 176061, India.

[2]Academy of Scientific and Innovative Research (AcSIR),

Ghaziabad, Uttar Pradesh, 201002, India.

Authors' contributions:

Ritu and SG carried out the study. NKS assisted in the study. RS conceptualized, designed, analyzed and supervised the entire study. Ritu, SG, NKS, and RS wrote the MS.

[†] Contributed Equally

[*] Corresponding Author: ravish@ihbt.res.in

Abstracts

Various noncoding elements of genome have gained attention for their regulatory roles where the lncRNAs are very recent and most intriguing for their possible functions. Due to limited information about lncRNAs, their characterization remains a big challenge, especially in plants. Plant lncRNAs differ a lot from others even in the mode of transcription and display poor sequence conservation. Scarce resources exist to annotate for lncRNAs with satisfactory reliability. Here, we present a deep learning approach-based software, DeepPlnc, to accurately identify plant lncRNAs across the plant genomes. DeepPlnc, unlike most of the existing software, can even accurately annotate the incomplete length transcripts also which are very common in *de novo* assembled transcriptomes. It has incorporated a bi-modal architecture of Convolution Neural Nets while extracting information from the sequences of nucleotides and secondary structure representations for plant lncRNAs. DeepPlnc scored high on all the considered performance metrics while breaching the average accuracy of >95% when tested across different experimentally validated datasets. The software was comprehensively benchmarked against some of the recently published tools to identify the plant lncRNAs where it consistently outperformed all the compared tools for all the performance metrics and for all the considered benchmarking datasets. DeepPlnc will be an important resource for reference free identification and annotation of transcriptome and genome for lncRNAs in plants. DeepPlnc has been made freely available as a web-server at https://scbb.ihbt.res.in/DeepPlnc/. Besides this, a stand-alone version is also provided at GitHub at https://github.com/SCBB-LAB/DeepPlnc/.

Introduction

Approximately 80% of the genome is commonly transcribed into RNA, of which majority belongs to non-coding RNAs (ncRNAs) alone (Djebali et al., 2012). More than 80% of these ncRNAs belong to long non-coding RNAs (lncRNAs) (Fok et al., 2017). Though not a clear definition or characterization process exists for them, the ncRNAs having length greater than 200 bases are considered as lncRNAs with short open reading frames (<100 amino acids) which are not translated into functional proteins (Kapranov et al., 2007). LncRNAs are supposed to act as a key modulator for various biological processes (Mercer et al., 2009). Their involvement is reported in controlling transcription process through enhancers and providing regulatory binding sites is well reported (Wang and Chekanova, 2017; Shankar R**,** 2020). They are also reported to work as sponges to miRNAs and suppress miRNA function by causing deflection in their supposed targeting (Wang et al., 2010). The lncRNAs are also found in nucleus and act as major components of nuclear speckles while being a partner in epigenetic and transcriptional controls (Hutchinson et al., 2007). In cytoplasm, lncRNAs interact with several RNA binding proteins (RBPs) and control their regulatory dynamics (Glisovic et al., 2008).

In plants, the first reporting of lncRNA was done in Soybean for ENOD40 (Yang et al., 1993). To this date, the number of experimentally identified lncRNAs in plants are much lesser than those reported for human and mouse, despite the fact that plant genomes are much more complex than animals' with several genomes much larger than the human genome. Unlike animal system based resources, there is a dearth of such resources for plants. The majority of the entries belong to the predicted ones through some rules based approached and lag behind enormously when compared to animals based resources and information. There are plant specific lncRNA databases like PLNlncRbase (Xuan et al., 2015), Plant long non-coding RNA database (PLncDB) (Jin et al., 2013), GreeNC (Gallart et al., 2016), Plant ncRNA Database (PNRD) (Yi et al., 2015) and CANTATAdb

(Szcześniak et al., 2016). Database like CANTATAdb contains 45,117 lncRNAs from 10 plant species with their coding potential and expression values . Most of the databases have redundant information, and majority have been derived from the predictions done by various tools like CNCI (Sun et al., 2013), CNIT (Guo et al., 2019), and CPC (Kong et al., 2007). Seeing such huge dependence of lncRNAs annotations in plants upon software alone itself makes a strong case for research on such tools. And when one looks for computational tools available to identify plant lncRNAs, more so for annotation and identification purpose, there is an enormous scope for improvement.

Compared to animals, plant lncRNAs differ even in their transcriptional modes and display much more complexity and diversity with marked poor degree of sequence conservation. This is one big reason for the evident scarcity of reliable tools and resources for plant lncRNA identification which mostly work with animal system identified properties. Some tools based on regular machine learning approaches like CPC (Kong et al., 2007) and CPC2 (Kang et al., 2017) have been developed where the inputs from human datasets were used to train the models. Yet, there are some recent tools like PLncPRO (Singh et al., 2017), RNAplonc (Negri et al., 2019), PreLnc (Cao et al., 2020), and CNIT (Guo et al., 2019) which have trained models on plant datasets exclusively. Coding Potential Calculator (CPC) and its upgraded version, CPC2, employ Support Vector Machine (SVM) models which depend upon four sequence intrinsic features (Fickett TESTCODE score, ORF length, ORF integrity, isoelectric point) to evaluate the protein-coding potential of transcripts, using ORF prediction quality from a BLASTX search based homology-based approach. PLncPRO is an another tool based on machine learning approach and applies random forest (RF) algorithm while using 71 features, including BLASTX outputs and frame entropy and frequencies of each of the 64 trimers (Singh et al., 2017). Another tool, RNAplonc, uses 16 features including sequence and structural free energy and applied REPTree machine learning algorithm (Negri et al., 2019). A more recently published tool, PreLnc, uses incremental feature selection method across the

traditionally identified features for lncRNAs. PreLnc is applying Pearson correlation co-efficient to reduce the total features and applies five different classifiers' comparisons (logistic regression (LR), SVM, decision tree (DT), RF, and K-nearest neighbor methods (KNN)) to build the models (Cao et al., 2020). In addition to these tools, CNIT, an upgraded version of CNCI tool, exhibits high accuracy using XGBoost machine learning algorithm. This tool considers both animal and plant datasets for model building. CNIT uses sequence based features such as maximum most-likely CDS score, their lengths, their standard deviation scores, and frequency of 64 codons (Guo et al., 2019).

Apart from the above mentioned machine learning approaches, which depend a lot upon successful manual identification and features extraction by the authors, very recently efforts have been made to involve deep learning methods to identify lncRNAs. In doing so, such deep learning networks extract important and even hidden transient features which would be otherwise hard to get through manual feature extraction approaches. Deep learning achieves this through intensifying the representation power of complex features by multifold (Wekasa et al., 2019). However, very limited development has happened in this direction for plant lncRNA identification. A tool, PlncRNA-HDeep, has been reported very recently which identifies plant lncRNAs applying two alternative deep learning models, a Long Short Term Memory (LSTM) and a Convolution Neural Net (CNN) framework (Meng et al., 2021) utilizing predicted dataset only for single species. Although, this tools does not provide any pre-bulit model for classification.

Despite all this progress, a fact remains that there is an enormous scope for improvement for plant specific lncRNA identification. First of all the existing machine learning approaches now need to move towards deep learning approaches where it would become possible to catch those features which are otherwise difficult to be caught through manual process of feature selection and identification. Traditional features like those based on k-mer and motifs based properties, length, ORF properties, and Kozak sequences may easily fail when sequence boundaries are not clear or

shorter incomplete transcripts are reported after assembling. Most of the existing tools report satisfactory scores when tested against some standard benchmarking dataset having full length sequences. Some of these datasets themselves need to be looked into as many instances of them are predicted instances which may lower the quality of training and negatively impact the learning process. As soon as these tools face the real situation where sequence boundaries are not known, be it genomic annotation or annotation of *de novo* assembled transcriptome, almost all of them fail, raising a big question mark on their actually applicability.

In the present study, we have developed a bi-modal CNN based deep learning model for lncRNAs identification. CNN have proved them tremendously successful in detecting spatial arrangements of patterns which are hard to get recognized otherwise. We expected CNN based deep-learning system to successfully detect such spatial arrangements of patterns in nucleotide sequences as well as structural sequences if they were transformed into pixeled form like images. Such approach more important in real case applications like annotation of genomes and *de-novo* assembled transcriptomes where boundaries are not clear and identification needs to be capable enough to work with incomplete sequences. Here, a base frame of 400 bases was used in an overlapping sliding window manner while keeping a cut-off of minimum 200 bases with padding. Under this frame, structural and sequence based properties were scanned directly in a parallel manner through two different CNN modules dedicated for each of them. Both of these CNNs were stacked upon two independent double layered fully connected dense feed forward neural nets. Output from these hidden dense layer from both the CNNs were finally concatenated on which the final classification was done. The entire system was developed by using experimentally validated lncRNA and known non-lncRNA instances and the system was tested across the different validated datasets, including the most commonly used dataset for the benchmarking of such tools. The developed tool, DeepPlnc, consistently performed high on the basis of performance metrics than the compared tools. The study also provides a very comprehensive benchmarking study done with some of the recently published

tools while giving actual picture of their performance. The study concludes with a practical application demonstration of the software on the *de novo* assembled transcriptome data of a Himalayan medicinal plant where it reported its lncRNAs for the first time.

Result and Discussion

Data collection and formation of datasets

A total of 9,911 experimentally validated plant ncRNAs sequences were downloaded from Ensembl Plants v51, PncStress, and PLncDB V2.0 databases for 51 plant species. Non-lncRNAs (mRNAs) data were downloaded from Ensembl Plants (v51) database for 51 plant species. The ncRNAs were used as positive data and the protein-coding transcript were used as negative data. This data contained 9,911 experimentally validated non-coding sequences. Sequences shorter than 200 nucleotides were discarded, keeping in the view that the minimum length of 200 bases is considered for lncRNAs. After filtration, this data got reduced to 6,563 sequences for 30 species. For non-lncRNA transcripts, 6,563 sequences were randomly obtained for the corresponding species considered for the lncRNA data. This dataset became the foundation data for training the deep learning modules. Initially, the above data was broken into two halves for training and testing datasets creation, containing 3,563 sequences non-lncRNAs and 3,563 sequences lncRNAs for training. Remaining 3,000 non-lncRNAs and 3,000 lncRNAs were used purely as the test samples. In this entire study this dataset is referred as Dataset “A”. Table 1 depicts the distribution of collected data across the various species in plants for Dataset “A”. This also needs to be mentioned here that this entire dataset split into training and testing sets has been done 10 times for 10 fold cross validation, building new model every time from the scratch and every time absolutely no overlap of instances between training and testing data, ensuring no training memory in order to get the unbiased performance measure.

Dataset “A” was also a neutral dataset where experimentally validated cases were considered and which was not used by any other tool considered here for comparison for model building. However, in order to keep the evaluation beyond the scope of any doubt, one more dataset was made. This dataset was built using the data sources like PNRD database and test datasets used in CNIT tool which were not used by the compared tools for their datasets and model building but followed the similar approach of annotating and classifying the lncRNAs. A total of 6,621 plant lncRNAs were obtained from PNRD database for 14 plant species and 439 lncRNAs were collected from the test set considered by CNIT for two plant species. A total of 7,060 lncRNA transcripts having minimum sequence length of 200 bases for 16 plant species were considered as the positive dataset. Non-lncRNAs data were downloaded from Ensembl Plants (v51) database for the corresponding 16 species considered for the lncRNA data. To generate balanced test set, 7,060 non-lncRNA sequences were randomly selected from non-lncRNAs dataset for each species which accounted for the negative dataset. In the lncRNA dataset (positive), majority of the lncRNAs (93.8%) were predicted by various computational methods where all RNAs were compared with the coding RNAs in RefSeq and Ensembl and later using CNIT these RNAs were filtered. Only the filtered RNAs were kept and considered as noncoding RNAs. For the rest of the sequences 6.2% of the total (7,060) for two plant species were reported as “known” sequences which were extracted from CNIT website (http://cnit.noncode.org/CNIT/) and did not overlap with Dataset “A” because it did not belong to the plant species which were considered for the Dataset “A”. This way another neutral dataset was created using lncRNAs and non-lncRNAs, referred as Dataset “B” in this entire study. Table 2 depicts the distribution of collected data across the various species in plants for Dataset “B”. The main purpose to use this dataset was to provide another blind test on a totally mutually exclusive dataset to enhance the confidence further, though Dataset “A” was also a totally neutral dataset and also contained experimentally validated cases. Also, this dataset was not applied for training purpose and was used exclusively for testing only.

Parameter optimization and establishment of the final Bi-modal CNN architecture

Figure 1 provides the details of the evolved CNN architecture to characterize the lncRNAs in plants. The framework of the bi-modal CNNs used in this study was mainly built by a 2D CNN layer, one max-pooling layer, and three fully connected layer with one output layer along with two batch normalization layers to control the over-fitting during the training process. We used bi-modal CNNs with two different parts which used data encoding by one hot encoding method into binary matrix as shown in Figure 1. In the one part of the bi-modal CNNs (left side of Figure 1A) the nucleotide sequence worked as the inputs. The input layer was followed by a convolution layer containing 32 channels at each position with 3x3 kernel size with sequence dimension of 400x4x1 (Input sequence length X Alphabets of the input X Input's depth). Exponential activation function was applied to each node of the channels set with a kernel constraint having maximum value three and padded with zero on both the sides of the input sequence in case the length was shorter than 400 bases in order to ensure a constant size of the input matrix. The output resulted into 400x4x32 dimension representation after convolution. This output was normalized by the second layer called batch normalization layer. This layer was applied with parameters like momentum = 0.99 and epsilon = 0.001 to overcome the over-fitting problem during training process. This layer was followed by a third layer called max-pooling layer. This layer included 32 nodes having kernel size 2x2 with input dimension 400x4x32. Max-pooling helped in reducing the dimensions of convoluted sequence into a dimension of 200x2x32 where the stride size was S=2. Next to this layer, pooled feature maps were passed to fully connected layer by flattening it into one dimensional form having an input vector of 12,800 nodes. In a fully connected layer, all the nodes of the previous hidden state were connected to every single node of the next hidden state layer. The hidden state layers in the present study had two dense layers with both having 32 hidden nodes with exponential activation function. The hidden layer output was considered as last return in the form of 32 dimension which was concatenated along with the output from the second CNN module as described below.

A similar CNN architecture was followed by the second part (right side of Figure 1A) of the bi-modal CNNs which considered secondary structural information in the form of dot bracket representation ("(", ".", ")") for the input sequence. The input layer passed on the input to the convolution layer having a dimension of 400x3x1 (Input sequence length X Alphabets in Dot-Bracket secondary structure representation X Input's depth) with 32 channels, 3x3 kernel sizes, stride size of one, and along with exponential activation functions with kernel constraints having maximum value 3. Inputs were padded with zero to ensure a constant size of the matrix in case of shorter input sequences. The second layer was max pooling which used normalized instances by passing through a batch normalization layer with input shape of 400x3x32 with stride size of two and gave output into reduced features maps with 200x1x32 dimension. This output became the input to a system of two hidden layers with fully connected nodes with 32 nodes per hidden layer with exponential activation function. It used an input size of 6,400 nodes.

At the end, the output from these two CNNs based networks were connected into one concatenated layer that received input in the form of 32+32 nodes, returning a vector of 64 elements. This vector was processed by one batch normalization layer and was passed into the last and final output layer that had one node with exponential activation function. Finally, the model was compiled by mean absolute error loss function to calculate the loss which was optimized by using the "FTRL" optimizer. The output of the last layer (output layer) returned the confidence probability for each input sequence while working on a window size of 400 bases (chunks). The confidence score indicated the confidence of each instance as non-lncRNA or lncRNA. Based on this confidence probability score (Cp), if the Cp is ≥ 0.50 means the corresponding input sequence was identified as lncRNA and if the Cp is < 0.50 then it is non-lncRNA for the chunk. For the full length sequences, if the average of the confidence probability of all the chunks for a particular sequences was found ≥ 0.50, the sequence was considered as lncRNA, otherwise as a non-lncRNA. The final parameters set

for the implemented model is as following: {"Activation function": Exponential, "Loss function": mean absolute error, "Optimizer": FTRL}. The related information are listed in Supplementary Table S1.

Ablation test for the properties effect on the network and performance assessment

Properties assessment is an important part in the training of a model. For the determination of the optimal feature combinations in the lncRNA classification model, we used six properties combinations for various input lengths for different input property encodings: 1) Monomers density, 2) Secondary structure, 3) Monomers density + Secondary-structure, 4) Dinucleotides density + Trinucleotides density, 5) Monomers density + Secondary-structure + Dinucleotides density + Trinucleotides density, 6) Monomers density + Secondary-structure + number of hydrogen bonds + purine-pyrimidine sequence encodings. Initially, the considered dataset was split in 50:50 ratio to form the training and testing datasets. Then, the model was trained on 200 bases long sequences and all the above mentioned properties encodings were done accordingly for that window size in sliding window manner. On the test set for the above mentioned input property encoding combinations, the accuracy values scored were 81.14%, 79.14%, 80.93%, 76.97%, 82.74%, and 80.08%, respectively. The window length was varied until the length of 800 bases and the scorings were obtained for all these window lengths. It was found that consistently best performance was achieved when the input window length was 400 bases long in overlapping manner. Even padded shorter sequences with shorter lengths as low as 200 bases performed the best for this window size. The average accuracy values scored for 400 bases window based classifiers were 83.57%, 83.78%, 92.01%, 85.45%, 86.19%, and 87.55%, respectively. These values were for the chunked datasets. A chunked dataset of 400 bases was generated from full length sequences dataset. Every chunk was associated with the same class tag to which the full length sequence belonged and from which the chunk was generated.

Figure 2(a) provides a snapshot of impact of the variable window sizes and combinations of property encodings considered for this chunked dataset. Finally, the combination of monomeric and dot-bracket secondary structure representations with a sliding window of 400 bases was used for the final model training which had scored an initial accuracy of 92.01% on the chunked dataset (87.95% sensitivity, 96.06% specificity, 91.25% F1-score, 0.84 MCC score). To take the final decision on the full length input sequence, all chunks were mapped back to the original input sequence and were assigned to the class lncRNA if average score of all the chunks were found higher than 0.5. In this way, the average accuracy of the above classifier for the complete length sequence increased and attained the accuracy value of 96.16% with 96.46% sensitivity, 95.86% specificity, 95.46% F1-score, 0.90 MCC score.

The next step was 10 fold cross validation to verify the consistency of the performance (Supplementary Table S2). Ten times the data was split in the ratio of 50:50 for training and testing datasets generation, and every time the corresponding model was built from the scratch and tested across the corresponding test set, following the same above mentioned approach. As can be found from the Figure 2(b), the developed system performed consistently good and scored high on all these training and testing combinations, where the average accuracy consistently scored above 92% and reached 96% for the complete length input sequences. As transpires from the scoring obtained for the various metrics, the model performed consistent and well balanced besides scoring high accuracy across the test data sets which were experimentally validated cases of non-lncRNAs and lncRNAs. The specific calculation results are shown in figure 2(b).

Benchmarking: DeepPlnc consistently outperformed all the compared tools

A highly comprehensive benchmarking study was performed where DeepPlnc was compared with five different recently published tools to detect plant lncRNAs: PlncPRO, CNIT, PreLnc, PlncRNA-HDeep, and RNAplonc. Besides this, the benchmarking also considered two different datasets (“A”,

“B”; already detailed above) to carry out a totally unbiased assessment of performance of these tools on different datasets. Each dataset was further divided into two sub-categories: chunked form containing incomplete RNA sequences of size 400 (shorter sequences were padded) and second full length sequences. This is important to note that chunked data also provided a frame to mimic the incomplete/truncated length sequences which usually appear in large number in the *de novo* assembled transcriptome data as well as provide a realistic treatment to deal with those conditions where boundaries of lncRNAs are not known, like genomic and *de novo* transcriptome annotations.

All these six software were tested across the two datasets (“A” and “B”) where DeepPlnc consistently outperformed all the compared tools for all these datasets, for both of their forms (incomplete chunked sequences and complete sequences), and for all of the performance metrics considered. Figure 3 gives a detailed view of the performance and benchmarking analysis across the two datasets studied for all these software. On the chunked transcript data representing the incomplete sequences, DeepPlnc scored 92.01% accuracy and 0.84 MCC scores, for Dataset “A”, taking a lead of ~4% in accuracy and ~0.07 lead in MCC scores from the second best performing tool, PLncPRO. On the complete length sequences, DeepPlnc scored the accuracy of 96.16% and the MCC value of 0.90 on Dataset “A”, taking a lead of ~4% for accuracy and ~0.05 lead for MCC while emerging as the best performer by larger margin. On the chunked dataset CNIT and PreLnc achieved an accuracy of 87.07% and 82.59% and MCC of 0.74 and 0.66, respectively, while for the complete sequences they achieved an accuracy of 87.95% and 92.60%, respectively. Their MCC values were 0.76 and 0.85, respectively. PlncRNA-HDeep and RNAplonc were the least performing tools on the chunked dataset, scoring an accuracy of 65.66% and 61.35%, respectively. Their MCC values were 0.36 and 0.36, respectively. However, RNAplonc achieved an accuracy of 92.20% and MCC of 0.84 on complete sequence dataset whereas PlncRNA-HDeep achieves an accuracy of 74.62% and MCC of 0.51. Figure 3 (a) and (b) provide the complete performance benchmarking for this Dataset “A” for chunked incomplete sequences and complete lengthed instances, respectively.

Coming to the Dataset “B”, DeepPlnc continued its top performance even on this dataset while scoring the highest values for all the performance metrics considered. It clocked an accuracy of 96.02% and the MCC value of 0.92 for this dataset on the complete length sequences. It scored 2.78% better accuracy than the second best performing tool, PLncPRO, and scored 0.05 higher MCC value than PLncPRO, on the complete length sequences. On the truncated chunked data of Dataset “B” also, DeepPlnc continued to outperform others with significant margin (Accuracy: 90.23%, MCC: 0.80) and maintained a margin of ~3% better accuracy than the second best performing tool, PLncPRO. It also scored ~0.03 better MCC score than PLncPRO. For the chunked dataset, CNIT and PreLnc achieved the accuracy scores of 77.36% and 72.19% and MCC values of 0.60 and 0.44, respectively. For the complete sequences they achieved an accuracy of 86.82% and 90.77% and MCC values of 0.73 and 0.81, respectively. PlncRNA-HDeep and RNAplonc were the least performing tools on the chunked dataset, scoring an accuracy of 73.83% and 56.25%, respectively. They scored MCC values of 0.48 and 0.26, respectively. However, RNAplonc achieved an accuracy of 92.63% and MCC 0.85 on the complete sequences. Whereas PlncRNA-HDeep achieved an accuracy of 79.07% and MCC value of 0.60. Figure 3 (c and d) provides the complete performance benchmarking for this Dataset “B” for the chunked incomplete sequences and complete length instances, respectively.

DeepPlnc scored an AUC value of 99.78% (incomplete sequences) and 99.24% (complete sequences) on Dataset “A”. For Dataset “B”, it scored the AUC values of 94.01% (incomplete chunked sequences) and 99.21% (complete sequences). As illustrated in Figure 4, after DeepPlnc, PLncPRO and PreLnc emerged as the next best AUC scorers. What is also noteworthy besides the high accuracy and F1-score values of DeepPlnc across all the datasets, is the fact that its MCC values always stood out with high margin than others. DeepPlnc maintained least variability and dispersion of performance scores and continued to display its strong balance in detecting the

positive and negative instances with high and similar level of precision. A higher MCC value is considered as the true reflector of the robustness and performance as it becomes higher only when all the four metrics of performance measure score high (Chicco and Jurman, 2020). It also means that higher the MCC value, better is the balance of the software between its sensitivity and specificity.

The benchmarking done here stands among one of the most comprehensive benchmarking done to evaluate the performances of the plant lncRNA identification tools. DeepPlnc consistently scored highest across all the comparative tests and clearly outperformed the compared tools. The full details and data for the benchmarking studies are given in Supplementary Table S3.

Application demonstration: Detection and annotation of novel lncRNAs in *Rheum australe* transcriptome data

To demonstrate the applicability of DeepPlnc, it was run against the recently published transcriptome data of an endangered Himalayan medicinal plant, *Rheum australe* whose genome sequence is still not known and absolutely no information is there for lncRNAs (Mala et al., 2021). The assembled transcriptome of *Rheum australe* had reported a total of 35,679 transcriptome sequences, of which 23,569 sequences were having significant BLAST hits. As mentioned by Mala et al., 2021, dissimilar sequence (DS) clustering was performed to identify the multiple representatives of the same gene which reduced the number of total unigenes from 23,569 to 21,303. These 21,303 unigene transcripts comprised well annotated protein coding RNAs (3,912), hypothetical (7,109) and predicted genes (10,282). Besides them, without any significant BLAST hit, a total of 12,110 transcripts stood as the potential candidates for lncRNA annotation. In this study, as also described in details in the methods section, for the sake of clarity we have kept aside the hypothetical/predicted gene categories. The well annotated non-lncRNAs were kept knowingly as it would provide the extent of error done on the prediction. A total of 2,991 out of 3,912 non-

lncRNAs with minimum length of 200 bases were selected to annotate the non-lncRNAs whereas 921 sequences were discarded due to shorter length. In similar way, a total of 6,598 transcripts were taken from 12,110 transcripts as input to DeepPlnc with minimum length of 200 bases to annotate the lncRNAs and rest of the 5,512 sequences were discarded. Figure 5 depicts the flow diagram representation of the process of lncRNA identification in *Rheum australe* and the result at its every step of execution. After this filtering process we had 2,991 transcripts as non-lncRNAs candidate and 6,598 transcript as potential lncRNAs candidate. A total of 5,889 transcripts out of 6,598 potential lncRNAs candidate were identified as lncRNAs and 709 transcript were identified as non-lncRNAs by DeepPlnc. Similarly, a total of 2,132 transcript were identified as non-lncRNAs out of 2,991 non-lncRNA candidate transcripts whereas 859 transcripts were wrongly classified by DeepPlnc. However, ~90% of the lncRNA transcripts were classified as the lncRNAs. In order to evaluate how other lncRNA identification tools characterized the data and how much wrong classifications they did to the well annotated non-lncRNA data, the two good performing tools from the benchmarking study described above, PLncPRO and CNIT, were also used to annotate this transcriptome data (Supplementary Table S4). CNIT classified 5,118 transcript as lncRNAs out of 6,598 potential lncRNAs and 1,480 transcripts as non-lncRNAs. Whereas, only 1,630 non-lncRNAs were correctly identified out of 2,991 non-lncRNAs candidate. Similarly, PLncPRO annotated 4,421 transcript as lncRNAs out of 6,598 potential lncRNAs and identified only 1,691 transcript as non-coding transcript out of 2,991 non-lncRNAs candidate. Figure 6 provides a Venn diagram of results obtained for this analysis and how these three tools performed on *Rheum australe* transcriptome annotation. A total of 6,748 and 2,841 transcripts were identified as lncRNA and non-lncRNA, respectively, through DeepPlnc. A total of 5,721 and 3,868 transcripts were identified as lncRNA and non-lncRNA, respectively, through PLncPRO. A total of 6,479 and 3,110 transcripts were identified as lncRNAs and non-lncRNAs, respectively, through CNIT. These values underlined the levels of confidence one could put on these tools when performing the annotation job. Also, it is possible that some of these transcripts annotated in the root study may not be really

coding in actual scenario and some a noncoding pseudogene transcripts as many of them are supposedly incomplete sequences.

Further delving on the transcriptome data was done for expression measures to assess how well the usual observation that lncRNAs are usually lesser expressed than non-lncRNAs. This was reflected by the identified lncRNAs in the transcripts of *Rheum australe*. The expression level which was measured in Fragments Per Kilobase of transcript per Million mapped reads (FPKM) of all the identified lncRNAs was compared with that of non-lncRNAs (Supplementary Table S5). The expression level of lncRNAs was found indeed lesser than non-lncRNAs at 4°C (t-test p-value 0.01). However, this gap in the expression was not that sharp at 25°C presumably due to lowering of the gene expression at higher temperature which is noticed in this Himalayan plant in response of higher temperature. The lncRNAs were found higher expressed at 4ºC than 25ºC (t-test p-value of 0.007). Since these tests were sensitive to some extreme values due to which mean values get influenced, medians/ranks based non-parametric ranking tests (Mann Whitney test) were also done. All these observation returned much higher significance, suggesting significant differences between lncRNAs expression and non-lncRNAs expression. Significantly higher expression of the lncRNAs was observed at the lower temperature for this plant. The mean and median expression values for lncRNAs at the lower temperature were 38.46 and 6.19, respectively. The same for non-lncRNAs were 39.89 and 7.01, respectively, at the lower temperature. At the higher temperature, the mean expression value was 19.73 and median expression value was 1.37 for the lncRNAs. While the non-lncRNAs displayed 18.56 mean expression value and 2.03 median expression value at the higher temperature.

For differential expression analysis, a total of 5,889 lncRNAs, as identified by DeepPlnc, were considered. In order to find the differentially expressed lncRNAs (DELs) at the two different temperatures (4°C and 25°C). A comparative analysis was performed for 4°C vs 25°C expression

status. Significantly up- and down-regulated DELs were identified using edgeR (Robinson et al., 2010) with log fold change (FC) ≥ 2 at a statistical significance level of $p \leq 0.05$ and false discovery rate (FDR) ≤ 0.05. A total 2,408 DELs were identified out of 5,889 lncRNAs, of which 674 were up-regulated and 1,734 were down-regulated at the lower temperature as compared to that at higher temperature (Supplementary Table S6). At 25°C, at total of 1,734 lncRNAs were found over-expressed, while at 4°C, at total of 674 lncRNAs were found over-expressed. Supplementary Table S6 contains the list of such differentially expressed lncRNAs. Furthermore, top 100 (50 each for up- and down-regulated) differentially expressed lncRNAs were searched against the nucleotide database of NCBI using BLASTN. No significant BLAST hit was found for most of them. For 28 such candidates for which some hit was found had coverage lesser than 30% of the sequence. A total of 72 were found with no significant BLAST hit was, 38 of which were over-expressed at the higher temperature and 34 were over-expressed at lower temperature. These 72 lncRNAs may be considered as novel lncRNAs of very high confidence (Supplementary Table S7). The identified lncRNAs in *Rheum* in response to temperature differences may be taken for molecular and experimental studies as they showed much sharper and significant responses against the temperature differences compared to the coding genes.

This all demonstrated the extent to which, tools like DeepPlnc could be important and useful in the annotation of totally new transcript datasets which contain a large number of incomplete sequences while giving much superior and reliable performance than previously published tools. Also, it underlines that how much importance lncRNAs hold in understanding the plant's response to environmental shift which appears much more sharper than the non-lncRNAs.

Webserver implementation

DeepPlnc has been made freely available a very simple to use webserver as well as a standalone program. The user needs to paste the RNA sequences in FASTA format into the text box or upload

the RNA sequences in FASTA format and then click the submit button for the identification. This data is run through the trained bi-modal CNN which generates a relative probability scores for every position in the sequences. Later, the result page appears from where results can be downloaded in a tabular format and sequence wise distribution of probability score in the form of line plot as well as violin plot is displayed on the result page in an interactive form. The server has been implemented using D3JS visualization library, Python, JavaScript, PHP, and HTML5. Figure 7 provides an overview of DeepPlnc webserver implementation. Besides this, a stand-alone version is also provided at GitHub at https://github.com/SCBB-LAB/DeepPlnc/.

Conclusion

The lncRNAs have emerged among important components of the eukaryotic transcriptome and regulatory systems. A growing number of lncRNAs are now being found to be implicated in plant growth, development, and maintenance. There have been some tools developed for plants to identify lncRNAs but their performance gaps leave ample space to develop new approaches. Most of the existing tools are shallow learning based and display a lot of inconsistency, data memory, species specific performance, and imbalance between sensitivity and specificity. Development with state-of-the-art approaches like deep learning remained an under-explored area in plant lncRNA biology. The present work provides one of the first deep learning approaches which returns high performance scores for all the considered metrics across a large volume of experimentally validated test datasets. It outperformed most recent and prominent tools for plant lncRNA identification with high consistency and balance, when tested across the different datasets while undertaking a very comprehensive benchmarking analysis. Applying bi-modal CNN system on sequence and structural information of RNAs, the developed software, DeepPlnc, was also found very accurately detecting even the incomplete transcripts. Such results make it a very important tool for annotating *de novo* transcriptome data which are studded with incompletely assembled transcripts and highly prone towards wrong annotations. Its capability to work on boundary-less inputs makes its practical

application for highly reliable annotations of genomes and transcriptomes. DeepPlnc has been implemented as a user-friendly web-server available at website: https://scbb.ihbt.res.in/DeepPlnc/ as well as is also made available as standalone program at GitHub at https://github.com/SCBB-LAB/DeepPlnc/.

Materials and Methods

Datasets

In this study, we considered experimentally validated plant lncRNAs as positive instances and fully annotated non-lncRNA (mRNA) as the negative instances. The lncRNA dataset was collected from Ensembl Plants (v51) with known status, PncStress, and PLncDB V2.0 repositories which provided lncRNA data from 51 plant species. The negative examples, i.e. non-lncRNA (mRNA) were downloaded from Ensembl Plants (v51) for corresponding plant species considered for lncRNAs. On the basis of length, positive and negative datasets were filtered. After filtration, the data was split into train and test sets in the ratio of 50:50. The train datasets were used to build the model and the remaining 50% (test set) was kept untouched during the building of model and were used for performance testing and comparison of the built models. For the performance stability and consistency evaluation, 10-fold cross validation was performed. Every time, the dataset was split randomly into 50:50 ratio with the first part used to rebuild the model from the scratch and the second part used to test it.

To further evaluate the performance of the DeepPlnc, another dataset was constructed and named as Dataset “B”. The Dataset “B” was derived from 16 plant species. This dataset was based on the known and predicted plant lncRNAs. The predicted lncRNAs were found in the similar manner as was done in previously published studies (Kang et al., 2017, Sun et al., 2013). The source for this dataset was from CNIT which had collected lncRNAs with status as “known” from Ensembl plants whereas the predicted lncRNAs were downloaded from PNRD database which is a comprehensive

integrated web resource for ncRNAs (http://structuralbiology.cau.edu.cn/PNRD/). The negative instances (mRNAs) for this dataset were downloaded from Ensembl plants (v51). The sequences were randomly selected for the corresponding 16 species considered for the lncRNA data. All the datasets used in the present study have been provided at the DeepPlnc webserver and in Supplementary Table S8.

Features encoded by One hot encoding method

In CNN, the input instances are required to be presented as vectors or matrices of numerical values (Rodriguez et al., 2018). Here, we encoded each base in the sequence with “one-hot” encoding (“A”: [1, 0, 0, 0], “T”: [0, 1, 0, 0], “G”: [0, 0, 1, 0], “C”: [0, 0, 0, 1]. The length of the input vector was set at 400 bases. The sequences with length less than 400 nucleotides were padded for their empty columns by zero. Since each base was converted into a four-dimensional vector, each sequence was vectorized into a matrix with a dimension of 400X4. We also used the structural information of these RNA sequence which hold special importance in case of lncRNAs where structural conservation is more expected than sequence conservation. This information of structure for each sequence was obtained in the form of Dot-Bracket representation of the secondary structure of the RNA by using RNAfold of ViennaRNA Package v2.4.18 (Lorenz et al., 2011). We encoded the structure information for each base’s corresponding dot-bracket assignment in the sequence with “one-hot” encoding (“(”: [1, 0, 0], “.”: [0, 1, 0], “)”: [0, 0, 1]) with a vector dimension of 400X3 (Input length X Alphabet of Dot-Bracket representation). Here also, padding was done for the shorter input sequences. The one-hot encoded bases and structural information are thus independent of one another. Applying both separately in a bi-modal manner would map the structural information to the sequence information while building the network models. The one-hot encoded metrics were then used as the input for the multi model convolution neural network to train and build the models for lncRNAs identification.

Deep learning implementation

In the present study, a bi-modal CNN architecture based deep learning system was designed and implemented. With the optimized features combination on the most suitable sequence length, feature vectors were used to build the models for lncRNAs characterization. The input was in the form of sequence of nucleotides and sequence of RNA secondary structure in the form of dot-bracket with the three states ("(", ".", ")") which go through two separate yet connected CNNs, which join each other before the final dense layer for the classification purpose. Both these CNN modules transform the nucleotide and structural sequences into a grid like representation with one-hot encoding which are fluxed through the two modules separately. To evaluate the performance of CNNs, various number of hidden layers were tested and finally two hidden layers were selected in fully connected manner. The number of the nodes across both the dense hidden layers were tuned based on the number of filters used in the convolution layer. The presented architecture of the bi-modal CNN consisted of one convolution layer, one pooling layer, one batch normalization layer, and two dense layers in both the CNNs blocks. Their output was joined in a concatenated layer with one batch normalization layer and one output layer in fully connected manner. The one-hot encoded representation of the sequences of bases and corresponding secondary structure representation pass through a two-dimensional convolution layer separately, where 32 convolution kernels of size 3X3 with strides of single step scan the grid representation of the sequences. This is followed by a batch normalization layer. The batch normalization was used to solve the over-fitting problem and improve the speed, performance and stability of CNNs. The normalized vectors finally pass through a max pooling layer, where features are compressed to reduce the complexity and decrease the computational time and resources. The kernel size of 2x2 and a stride of two steps was fixed for this layer to retain the maximum value in the patch of each feature map. Max-pooling layer was followed by a flatten layer which converted the data into one dimensional array to create a single dimensional feature representation while holding the output from the previous steps as its input. The output from the flattening becomes the input to two hidden dense layers which helped in the final

classification step. At this stage the outputs of the dense layers from the two modules concatenate together and undergoes batch normalization. An exponential activation function based single node classification layer was used with Mean Absolute Error loss function to calculate the loss. “FTRL” optimizer was used at this point to adjust the weights and learning rates. FTRL is suitable for shallow models with large and sparse feature spaces. It applies different learning rates for different dimensions. The values of learning rate changed from 0.001 to 0.01. L1 and L2 regularization were also used where L1 regularization attempts to estimate the median of the data and L2 regularization makes estimation of the mean of the data in order to evade overfitting. The batch size was set to 100 and the number of epochs was set to 60. The DeepPlnc was implemented in Keras using TensorFlow. All codes were written for Python3. All the parametric values in the combination were tested and fixed using an in-house developed script to get the best parameter values combination. The final layer outputs a confidence probability score indicated the confidence of each instance as non-lncRNA or lncRNA. If confidence probability score (Cp) ≥ 0.50 means the corresponding input sequence was identified as lncRNA and if the Cp is < 0.50 then it is non-lncRNA for the chunk. For the full length sequences, if the average of the confidence probability of all the chunks for a particular sequences was ≥ 0.50, the sequence was considered as lncRNA, otherwise as a non-lncRNA.

Performance evaluation criteria

The performance of the developed model was evaluated on the test sets while applying the standard performance metrics. Confusion metrics containing correctly and incorrectly identified test set instances were built for the model. Sensitivity defines the portion of the positive instances which were correctly identified as positives whereas the specificity value describes the portion of negative instances which were correctly identified as negatives. Precision estimates the proportion of positives with respect to the total true and false positives. F1-score was also evaluated which measures the balance between precision and recall. Besides these metrics, Matthew’s Correlation

Coefficient (MCC) was also considered. MCC is considered among the best metrics to fathom the performance where score is equally influenced by all the four confusion matrix classes (true positives, false negatives, true negatives, and false positives) (Chicco and Jurman, 2020). Better the MCC score, the robust and balanced model with high degree of performance consistency.

Performance measures were done using the following equations:

$$\text{Acc}=\frac{\text{TN}+\text{TP}}{(\text{TN}+\text{TP}+\text{FN}+\text{FP})}$$

$$\text{Specificity}(\text{Sp})=\frac{\text{TN}}{(\text{TN}+\text{FP})}$$

$$\text{Precision}=\frac{\text{TP}}{(\text{TP}+\text{FP})}$$

$$\text{Recall}/\text{Sensitivity}(\text{Sn})=\frac{\text{TP}}{(\text{TP}+\text{FN})}$$

$$F1-\text{Score}=2\times\left(\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\right)$$

$$\text{AUC}=\int_0^1 \text{Pr}[\text{TP}](v)\,\text{dv}$$

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

Where:

True Positive (TP) indicates the number of the lncRNAs which are correctly identified, True Negatives (TN) indicates the number of true non-lncRNAs which are correctly identified, False Positives (FP) indicates the number of true non-lncRNAs which are incorrectly identified as lncRNAs, False Negatives (FN) indicates the number of true lncRNAs which are incorrectly identified as non-lncRNAs. Accuracy (Acc) is the percentage of the correctly identified data samples out of the total samples. In addition to it, Area Under Curve (AUC) from receiver operating

characteristic (ROC) curves were also used for evaluation. The value of AUC ranges from 0 to 1, where AUC = 1 stands for the perfect prediction. F1-score is a harmonic average of sensitivity and precision. MCC indicates a correlation coefficient between the true classes and the predicted classes.

Benchmarking and performance evaluation

To evaluate the DeepPlnc performance in this study, we compared DeepPlnc with five different tools: PlncPRO (Singh et al, 2017), CNIT (Guo et al., 2019), PreLnc (Cao et al., 2020), PlncRNA-HDeep (Meng et al., 2021), and RNAplonc (Negri et al., 2019). Two different datasets were considered separately for the benchmarking process as already described above (Dataset “A” and Dataset “B”). As already mentioned above, DeepPlnc dataset was based on the positive datasets from experimentally validated lncRNA sequences retrieved from PncStress, PLncDB v2.0, and Ensembl Plants (v51). The negative datasets were based on the protocol mentioned above in the previous section. The Dataset “B” was based on the positive datasets from both predicted and known lncRNA sequences as described above.

The four out of the compared five tools viz. PLncPRO, CNIT, PreLnc, and RNAplonc provide pre-built models. Only PlncRNA-HDeep does not provide any pre-built models. To overcome this problem, we built the model as described in its publication (Meng et al., 2021). Both PLncPRO and PreLnc provide two different models for lncRNA classification. For PLncPRO, monocot and dicot are the two models available whereas for PreLnc, model built from *Zea mays* and *Arabidopsis thaliana* were available for prediction. In DeepPlnc, the sequence was scanned by an overlapping window of 400 bases at a time. For the identification with DeepPlnc, if more than 50% of the 400 bases long overlapping windows scored higher value for lncRNA ($Cp \geq 0.50$) then the sequence was labeled as lncRNA, otherwise non-lncRNA. The sequence having length < 200 bases were

discarded as they did not fit one of the formally accepted prime criteria to define lncRNAs. Both the considered datasets were tested in the form of full length complete sequences sets as well as randomly chopped sequences to mimic the incomplete sequences.

Annotation of lncRNAs in *Rheum australe* and differential expression analysis

An endangered medicinal herb of high-altitude alpine region of Himalayas *Rheum australe* is known to possess anti-cancerous properties. Temperature is one of the major environmental factors in the niche of *Rheum australe* which determines its growth and development. To compare and understand the influence of temperature in terms of lncRNA transcripts response, the transcriptome profiling of the plant samples at 4°C and 25°C were retrieved studies (Mala et al., 2021). The study provided a total of 21,303 unigenes. These unigenes were further grouped according to their known annotations: 1) fully annotated proteins, 2) predicted proteins, 3) Hypothetical proteins and these three were considered as protein-coding genes. A total of 12,110 transcripts without any siginificant BLAST hits were considered as the potential lncRNAs candidates. Furthermore, sequences having length greater than or equal to 200 nucleotides were considered for the further study and were used for the identification of lncRNAs. As implemented in DeepPlnc, sequences having average confidence probability ≥ 0.50 were classified as lncRNAs and if the average confidence probability was < 0.50 the sequence was classified as a non-lncRNA. Similarly, we selected PLncPRO and CNIT tools to check their performance on this transcriptomic dataset with default parameters. Before going into further analysis, we removed the predicted and hypothetical protein coding sequences for the study because there was no experimental evidence that if such sequences really coded proteins or not and fully annotated protein coding sequences. LncRNAs identified by DeepPlnc tool were kept for further differential expression analysis, while the whole transcriptome pool defined above for containing lncRNAs and non-lncRNAs were retained to make comparative analysis for the performance of the compared tools on this data.

To analyze the changes in expression levels of the coding RNAs and lncRNAs at the two temperatures (4°C and 25°C), expression levels of all the sequences were calculated by RSEM software in terms of FPKM values. The statistical significance of differences in the expression levels (FPKM values) of lncRNAs for the two conditions (4°C and 25°C) were calculated using "t-test" in python and Mann-Whitney test in R.

The lncRNAs (confidence probability ≥ 0.50) and non-lncRNAs (confidence probability < 0.50) identified in *Rheum australe* were analyzed for differential expression via edgeR package (Robinson et al., 2010). In order to find the differential expressed lncRNAs (DELs) among 4°C and 25°C, a comparative analysis was performed for lncRNAs to identify significantly up- and down-regulated DELs with $\log_2$ FC ≥ |2| at a statistical significance level of $p \leq 0.05$ and FDR ≤ 0.05. After this, we picked up up-regulated lncRNAs from overall predicted lncRNAs in both conditions, followed by homology search against nucleotide database using BLASTN to look for any possible annotation for the identified novel lncRNAs. The lncRNAs having BLAST hits were considered as pre-reported and annotated ones for protein coding regions and remaining lncRNAs with no BLAST hits were studied for the potential novel lncRNAs.

Declarations

Availability of data and materials

All the secondary data used in this study were publicly available and their due references and sources have been provided. All data and information generated/used, methodology related details etc. have also been made available in the supplementary data files provided along with and also made available through the related open access server at https://scbb.ihbt.res.in/DeepPlnc/.

Competing interests

The authors declare that they have no competing interests.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Supplemental information

Supplementary Table S1: Hyperparameter optimization for DeepPlnc.

Supplementary Table S2: Ten fold cross validation of trained bi-modal CNN model.

Supplementary Table S3: Benchmark performance of DeepPlnc and other selected tools on two diiferent datasets.

Supplementary Table S4: Classification performance of tools on *Rheum australe* transcriptome.

Supplementary Table S5: FPKM values of lncRNAs detected by DeepPlnc in *Rheum australe.*

Supplementary Table S6: Diffrentially expressed lncRNAs at 4 °C vs 25 °C.

Supplementary Table S7: Top 50 differentially expressed lncRNAs for BLAST search in each condition.

Supplementary Table S8: Source information of Dataset “A” and Dataset “B” used to construct, evaluate and benchmark the DeepPlnc bi-modal CNN.

Funding information

RS is thankful to Council of Scientific and Industrial Research, Govt. of India for supporting this study through grant in iPRESS [Grant number: 34/1/TD-AgriNutriBiotech/NCP-FBR 2020-RPPBDD-TMD-SeMI(MLP-0156)] to RS.

Acknowledgments

The study was carried out under the aegis of The Himalayan Centre for High-throughput Computational Biology (HiCHiCoB), a BIC supported by the Dept. of Biotechnology, Govt. Of India. We are thankful to CSIR for the funding support they gave for this project (iPRESS). Ritu is thankful to CSIR for her fellowship and to Academy of Scientific and Innovative Research (AcSIR) for her Ph.D. enrollment. SG is thankful to CSIR project iPRESS for his project associateship. NKS is thankful to CSIR project iPRESS for his senior project associateship. All authors are thankful to the Director, CSIR-IHBT, for his kind support for this study.

Tables

Table 1: Table for experimentally validated lncRNAs which were used to construct and evaluate the DeepPlnc multimodal CNN. The table depicts the distribution of collected data across the various species in plants.

S.No.	Species	Count
1	Aegilops tauschii	12
2	Amborella trichopoda	4
3	Arabidopsis thaliana	3,856
4	Beta vulgaris	3
5	Brassica napus	7
6	Brassica oleracea	3
7	Brassica rapa	1
8	Citrus clementina	1

9	Digitalis purpurea	29
10	Glycine max	1
11	Gossypium hirsutum	11
12	Helianthus annuus	3
13	Hordeum vulgare	19
14	Morus multicaulis	1
15	Medicago truncatula	2
16	Musa acuminata	1
17	Nicotiana attenuata	2
18	Oryza glaberrima	3
19	Oryza longistaminata	2
20	Oryza meridionalis	1
21	Oryza rufipogon	1
22	Oryza sativa	1
23	Populus simonii	41
24	Solanum tuberosum	1
25	Triticum aestivum	28
26	Triticum dicoccoide	13
27	Triticum urartu	7
28	Vigna radiata	1
29	Vitis vinifera	14
30	Zea mays	2,494

Table 2: Species wise composition of Dataset “B”. The dataset contains known as well as predicted lncRNAs which were used to evaluate the DeepPlnc and the compared tools.

S.No.	Species	Count
1	Arabidopsis lyrata	477
2	Arabidopsis thaliana	2,622
3	Brassica rapa	1
4	Cucumis sativus	7
5	Gossypium barbadense	1
6	Hordeum vulgare	1
7	Medicago truncatula	2
8	Oryza brachyantha	418

9	Oryza sativa	790
10	Populus trichocarpa	564
11	Setaria italica	8
12	Solanum lycopersicum	8
13	Theobroma cacao	21
14	Triticum aestivum	1
15	Vitis vinifera	1
16	Zea mays	2,138

Figure legends

Figure 1: Detailed pipeline of the workflow. A) The image provides the brief outline of entire computational protocol implemented to convert full length sequences into 400 nts length sequences that were further processed for feature generation such as nucleotide sequence and secondary structure derived from RNAfold tool for each block respectively and used as input features for bi-modal convolution neural network. B) Architecture of bi-modal CNN. C) Visualization of bi-modal CNN having two blocks, one Convolution Neural Network and second Fully connected layers with input and output dimensions of each layers.

Figure 2: a) Accuracy plot for different feature combinations used to select important features. Third combination which was monomer + secondary structure reported higher accuracy than other combinations [1-kmer(monomers), Structure(Secondary structure), 2-kmer (Dinucleotides), 3-kmer (Trinucleotides), H-bond (Hydrogen bond), Pu-py (Purine-Pyrimidine)]. **b)** Ten-fold cross validation of DeepPlnc on Dataset “A” depicts consistent performance of the tool on randomly shuffled data to train and evaluate the model.

Figure 3: Comparative benchmarking with different combinations of test datasets. **(a)** Chunk Dataset “A”, **(b)** Full-length Dataset “A”, **(c)** Chunk Dataset “B”, and **(d)** Full-length Dataset “B”. For every such combinations, the performance metrics (sensitivity, specifcity, F1-score, MCC and

accuracy) are given in the form of heatmap. The plots clearly indicate that DeepPlnc consistently outperforms in comparison to existing tools for all the metrics based on all these different combinations on different datasets. Consistently high MCC scoring obtained by DeepPlnc reflects it as a robust tool where dispersion in performance metrics was least and DeepPlnc also scored high sensitivity for all set of combinations, clearly depicts that it performs better than other plant lncRNA classifier tool. MCC values were converted to percentage representation for scaling purpose.

Figure 4: ROC curve plot obtained by DeepPlnc and all others tools for all combinations, **(a)** Chunk Dataset "A", **(b)** Full-length Dataset "A", **(c)** Chunk Dataset "B" , and **(d)** Full length Dataset "B".

Figure 5: Flow diagram representing the process of lncRNA identification in *Rheum australe*. Compared to other tools, performance of DeepPlnc was found much superior. Its wrong classification for the well annotated RNAs as lncRNA was the least. Also, it classified most of the potential lncRNAs correctly where also it performed better than others.

Figure 6: a) Venn diagram showing the number of coding and long non-coding transcripts annotated by CNIT, PLncPRO, and DeepPlnc on coding dataset. **b)** Venn diagram showing the number of coding and long non-coding transcripts annotated by CNIT, PLncPRO, and DeepPlnc on the long non-coding dataset. Common entities in two or more tools are enclosed in the overlapping portion of the ellipses.

Figure 7: DeepPlnc webserver implementation. **A)** Input data box where the user can either paste or load the input file, **B)** Input is RNA sequence in FASTA format, **C)** These sequences were chopped into sequences of length 400 base pair in overlapping fashion and their secondary structure were

obtained by running RNAfold. Later, sequence and their secondary structure represented by triplets were converted into one hot representation which was finally fed into the bi-modal CNN which in turn gives the position wise probability scores, **D)** These probability scores are represented in the form of interactive violin plot depicting the distribution of probability score on the ten different sequence for the selected batch, **E)** Also, the probability scores are represented in the form of interactive line plot depicting the distribution of probability score across position for the selected sequence. **F)** Download option for the result in the tabular format where the first column indicates sequence id and second column represents whether it is lncRNA or not.

Reference

Cao L, Wang Y, Bi C, Ye Q, Yin T, Ye N (2020) PreLnc: An Accurate Tool for Predicting lncRNAs Based on Multiple Features. Genes (Basel) 11: E981

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21: 6

Djebali S, Davis CA, Merkel A, Dobin A, Lassmann T, Mortazavi A, Tanzer A, Lagarde J, Lin W, Schlesinger F, et al (2012) Landscape of transcription in human cells. Nature 489: 101–108

Fok ET, Scholefield J, Fanucchi S, Mhlanga MM (2017) The emerging molecular biology toolbox for the study of long noncoding RNA biology. Epigenomics 9: 1317–1327

Glisovic T, Bachorik JL, Yong J, Dreyfuss G (2008) RNA-binding proteins and post-transcriptional gene regulation. FEBS Lett 582: 1977–1986

Guo J-C, Fang S-S, Wu Y, Zhang J-H, Chen Y, Liu J, Wu B, Wu J-R, Li E-M, Xu L-Y, et al (2019) CNIT: a fast and accurate web tool for identifying protein-coding and long non-coding transcripts based on intrinsic sequence composition. Nucleic Acids Res 47: W516–W522

Hutchinson JN, Ensminger AW, Clemson CM, Lynch CR, Lawrence JB, Chess A (2007) A screen for nuclear transcripts identifies two linked noncoding RNAs associated with SC35 splicing domains. BMC Genomics 8: 39

Jin J, Liu J, Wang H, Wong L, Chua N-H (2013) PLncDB: plant long non-coding RNA database. Bioinformatics 29: 1068–1071

Kang Y-J, Yang D-C, Kong L, Hou M, Meng Y-Q, Wei L, Gao G (2017) CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features. Nucleic Acids Res 45: W12–W16

Kapranov P, Cheng J, Dike S, Nix DA, Duttagupta R, Willingham AT, Stadler PF, Hertel J, Hackermüller J, Hofacker IL, et al (2007) RNA maps reveal new RNA classes and a possible function for pervasive transcription. Science 316: 1484–1488

Kong L, Zhang Y, Ye Z-Q, Liu X-Q, Zhao S-Q, Wei L, Gao G (2007) CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. Nucleic Acids Res 35: W345-349

Lorenz R, Bernhart SH, Höner Zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL (2011) ViennaRNA Package 2.0. Algorithms Mol Biol 6: 26

Mala D, Awasthi S, Sharma NK, Swarnkar MK, Shankar R, Kumar S (2021) Comparative transcriptome analysis of Rheum australe, an endangered medicinal herb, growing in its natural habitat and those grown in controlled growth chambers. Sci Rep 11: 3702

Meng J, Kang Q, Chang Z, Luan Y (2021) PlncRNA-HDeep: plant long noncoding RNA prediction using hybrid deep learning based on two encoding styles. BMC Bioinformatics 22: 242

Mercer TR, Dinger ME, Mattick JS (2009) Long non-coding RNAs: insights into functions. Nat Rev Genet 10: 155–159

Negri T da C, Alves WAL, Bugatti PH, Saito PTM, Domingues DS, Paschoal AR (2019) Pattern recognition analysis on long noncoding RNAs: a tool for prediction in plants. Brief Bioinform 20: 682–689

Paytuví Gallart A, Hermoso Pulido A, Anzar Martínez de Lagrán I, Sanseverino W, Aiese Cigliano R (2016) GREENC: a Wiki-based database of plant lncRNAs. Nucleic Acids Res 44: D1161-1166

Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26: 139–140

Rodríguez P, Bautista MA, Gonzàlez J, Escalera S (2018) Beyond One-hot Encoding: lower dimensional target embedding. Image and Vision Computing 75: 21–31

Shankar R (2020) Chapter 4 - The dynamic aspects of RNA regulation. In R Pandey, ed, Rna-Based Regulation in Human Health and Disease. Academic Press, pp 85–115

Singh U, Khemka N, Rajkumar MS, Garg R, Jain M (2017) 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 45: e183

Sun L, Luo H, Bu D, Zhao G, Yu K, Zhang C, Liu Y, Chen R, Zhao Y (2013) Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. Nucleic Acids Res 41: e166

Szcześniak MW, Rosikiewicz W, Makałowska I (2016) CANTATAdb: A Collection of Plant Long Non-Coding RNAs. Plant Cell Physiol 57: e8

Wang H-LV, Chekanova JA (2017) Long Noncoding RNAs in Plants. Adv Exp Med Biol 1008: 133–154

Wang J, Liu X, Wu H, Ni P, Gu Z, Qiao Y, Chen N, Sun F, Fan Q (2010) CREB up-regulates long non-coding RNA, HULC expression through interaction with microRNA-372 in liver cancer. Nucleic Acids Res 38: 5366–5383

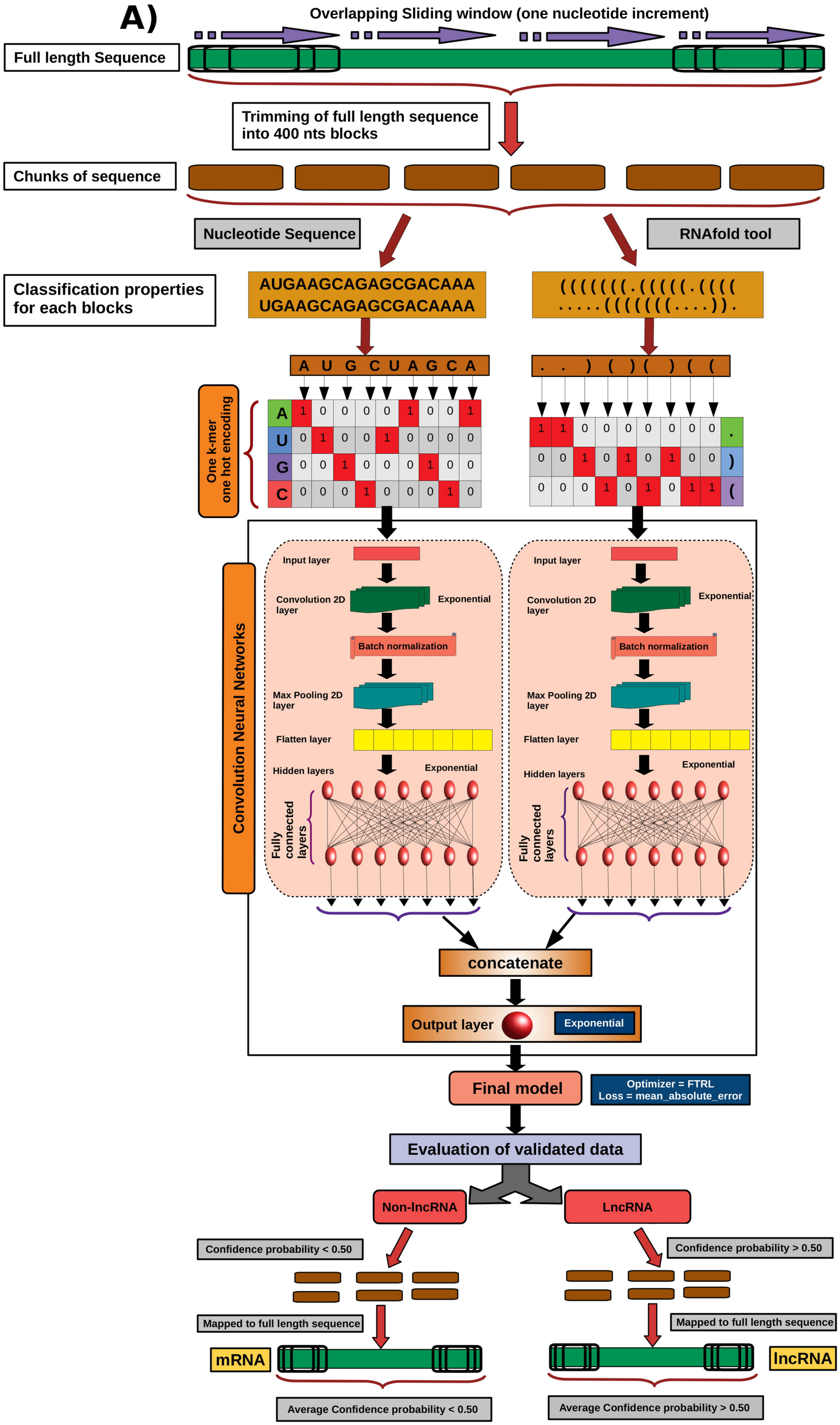
Wekesa JS, Luan Y, Chen M, Meng J (2019) A Hybrid Prediction Method for Plant lncRNA-Protein Interaction. Cells 8: E521

Xie C, Yuan J, Li H, Li M, Zhao G, Bu D, Zhu W, Wu W, Chen R, Zhao Y (2014) NONCODEv4: exploring the world of long non-coding RNA genes. Nucleic Acids Res 42: D98-103

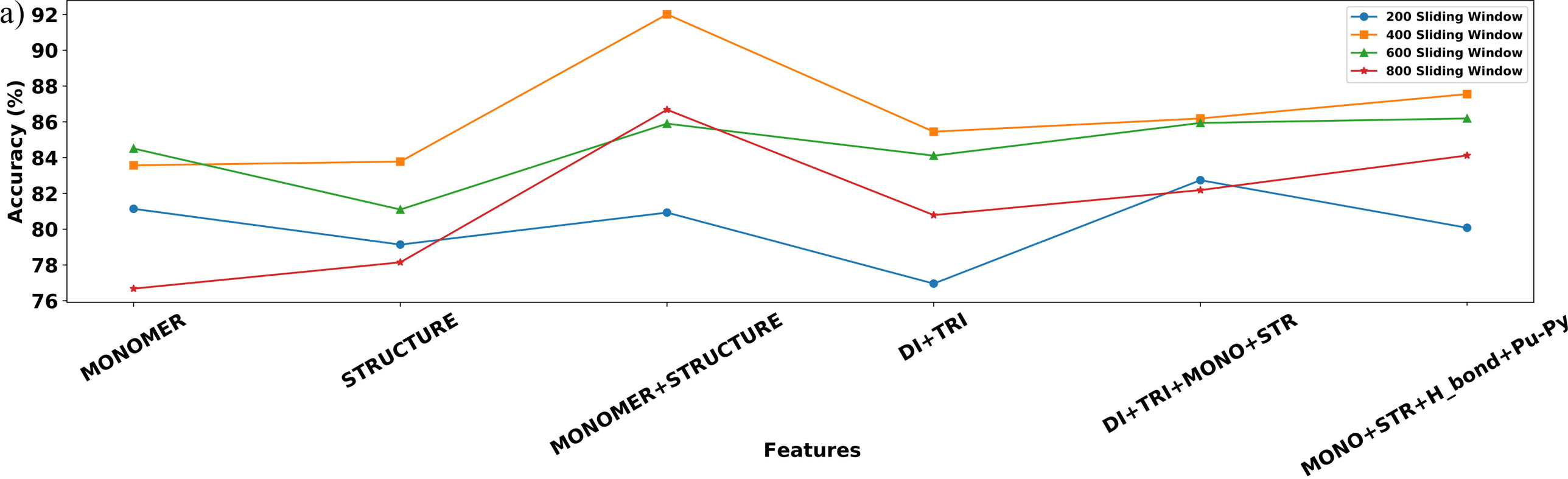
Xuan H, Zhang L, Liu X, Han G, Li J, Li X, Liu A, Liao M, Zhang S (2015) PLNlncRbase: A resource for experimentally identified lncRNAs in plants. Gene 573: 328–332

Yang WC, Katinakis P, Hendriks P, Smolders A, de Vries F, Spee J, van Kammen A, Bisseling T, Franssen H (1993) Characterization of GmENOD40, a gene showing novel patterns of cell-specific expression during soybean nodule development. Plant J 3: 573–585

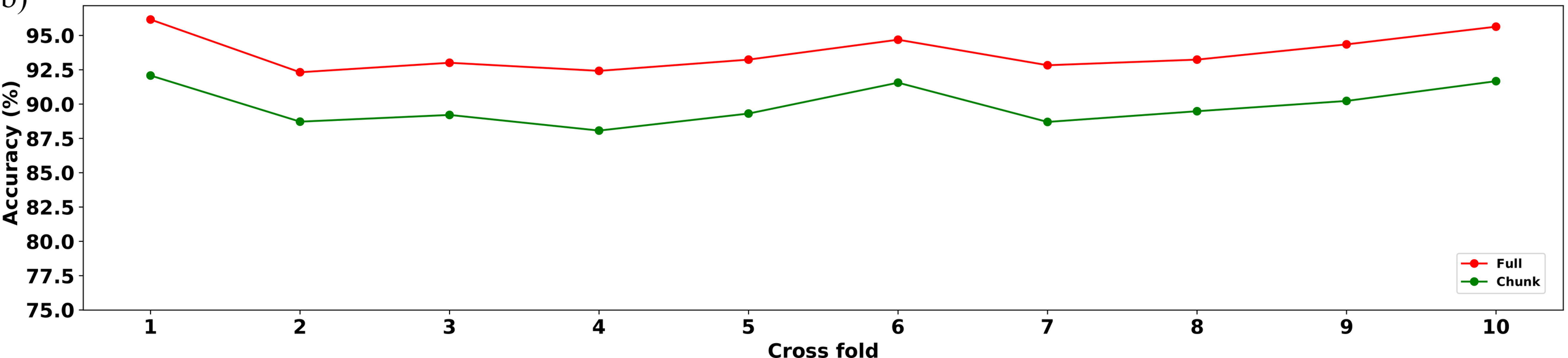
Yi X, Zhang Z, Ling Y, Xu W, Su Z (2015) PNRD: a plant non-coding RNA database. Nucleic Acids Res 43: D982-989

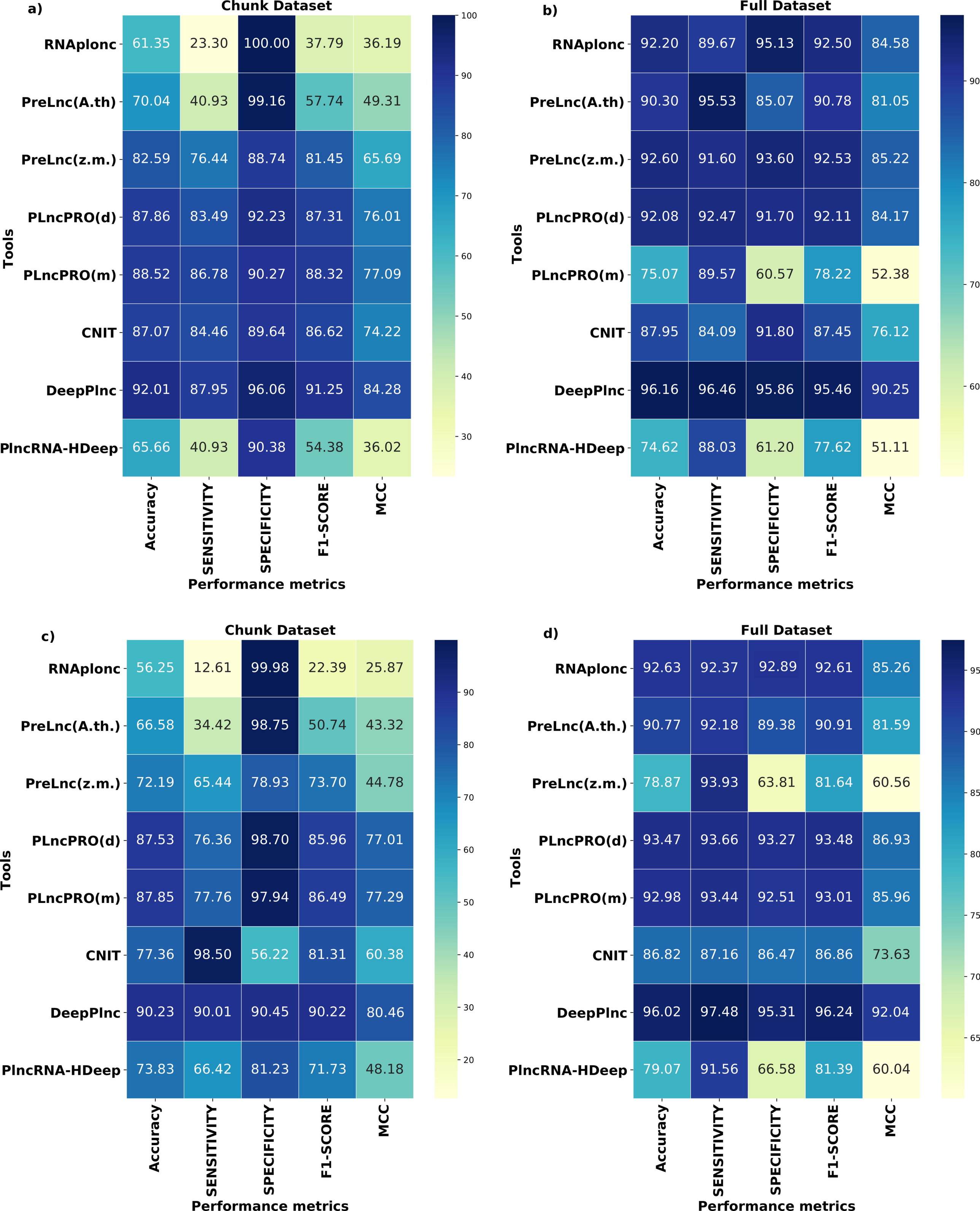

A)
Overlapping Sliding window (one nucleotide increment)
Full length Sequence
Trimming of full length sequence into 400 nts blocks
Chunks of sequence
Nucleotide Sequence
RNAfold tool
Classification properties for each blocks
AUGAAGCAGAGCGACAAA
UGAAGCAGAGCGACAAAA
(((((((.(((((.((((
.....(((((((....)).
A U G C U A G C A
. .) ( ) ( ) ((
One k-mer one hot encoding
Convolution Neural Networks
Input layer
Convolution 2D layer
Exponential
Batch normalization
Max Pooling 2D layer
Flatten layer
Hidden layers
Exponential
Fully connected layers
concatenate
Output layer
Exponential
Final model
Optimizer = FTRL
Loss = mean_absolute_error
Evaluation of validated data
Non-lncRNA
LncRNA
Confidence probability < 0.50
Confidence probability > 0.50
Mapped to full length sequence
mRNA
lncRNA
Average Confidence probability < 0.50
Average Confidence probability > 0.50
B)
Sequence Dataset
Secondary structure Dataset
One hot encoding
Input (400)
Output (400, 4,1)
Output (400, 3,1)
Convolution Neural Network
Conv2D-layer
Input (400, 4, 1)
Output (400, 4, 32)
Input (400, 3, 1)
Output (400, 3, 32)
Batch Normalization-1
Input (400, 4, 32)
Output (400, 4, 32)
Input (400, 3, 32)
Output (400, 3, 32)
Max-pooling layer
Input (400, 4, 32)
Output (200, 2, 32)
Input (400, 3, 32)
Output (200, 1, 32)
Flatten layer
Input (200, 2, 32)
Output (12800)
Input (200, 1, 32)
Output (6400)
Fully connected Layer
Dense layer-1
Input (12800)
Output (32)
Input (6400)
Output (32)
Dense layer-2
Input (32)
Output (32)
Input (32)
Output (32)
Concatenate
Input (32),(32)
Output (64)
Batch Normalization-2
Input (64)
Output (64)
Exponential
Dense layer-3
Input (64)
Output (1)

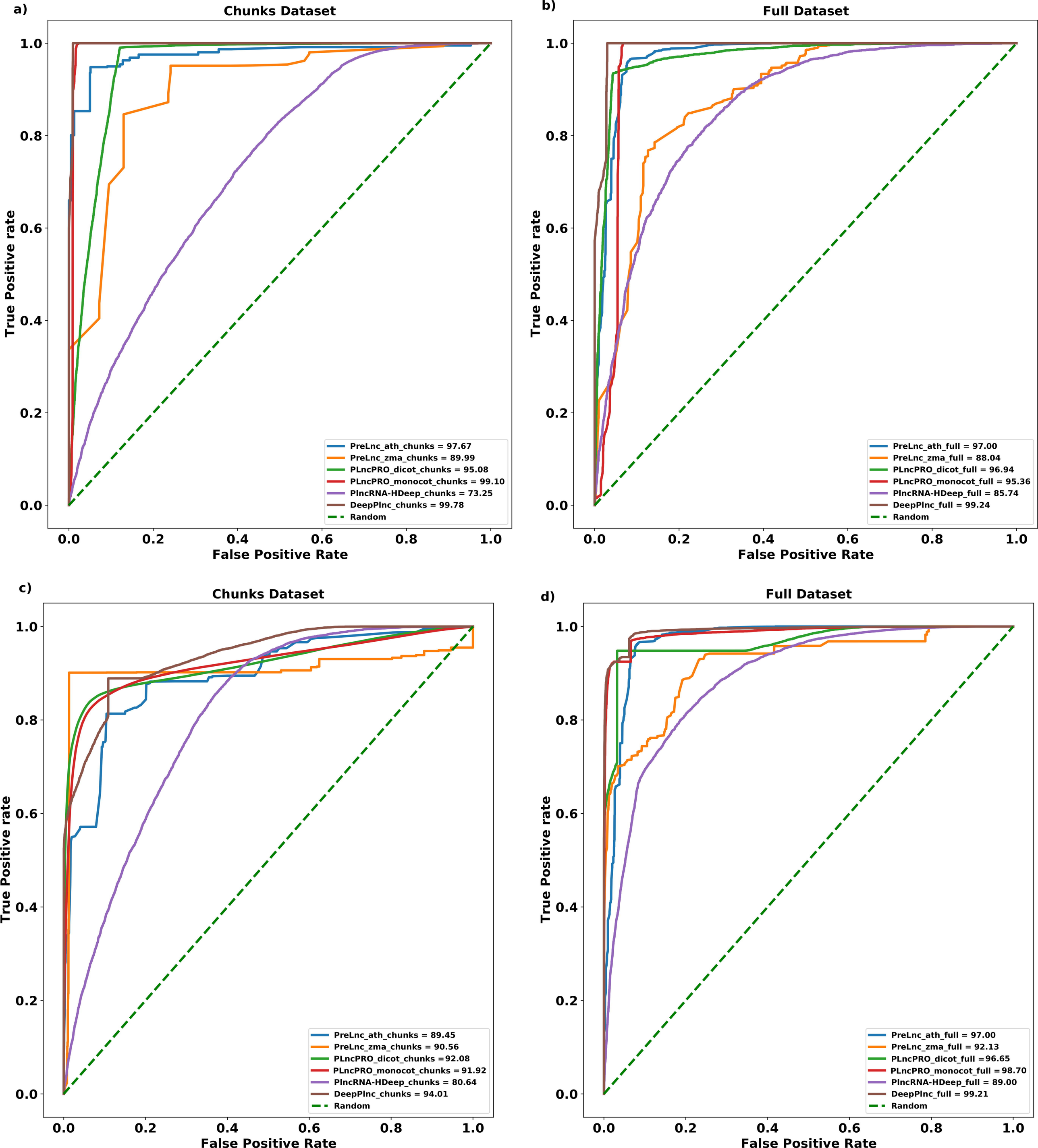
a)

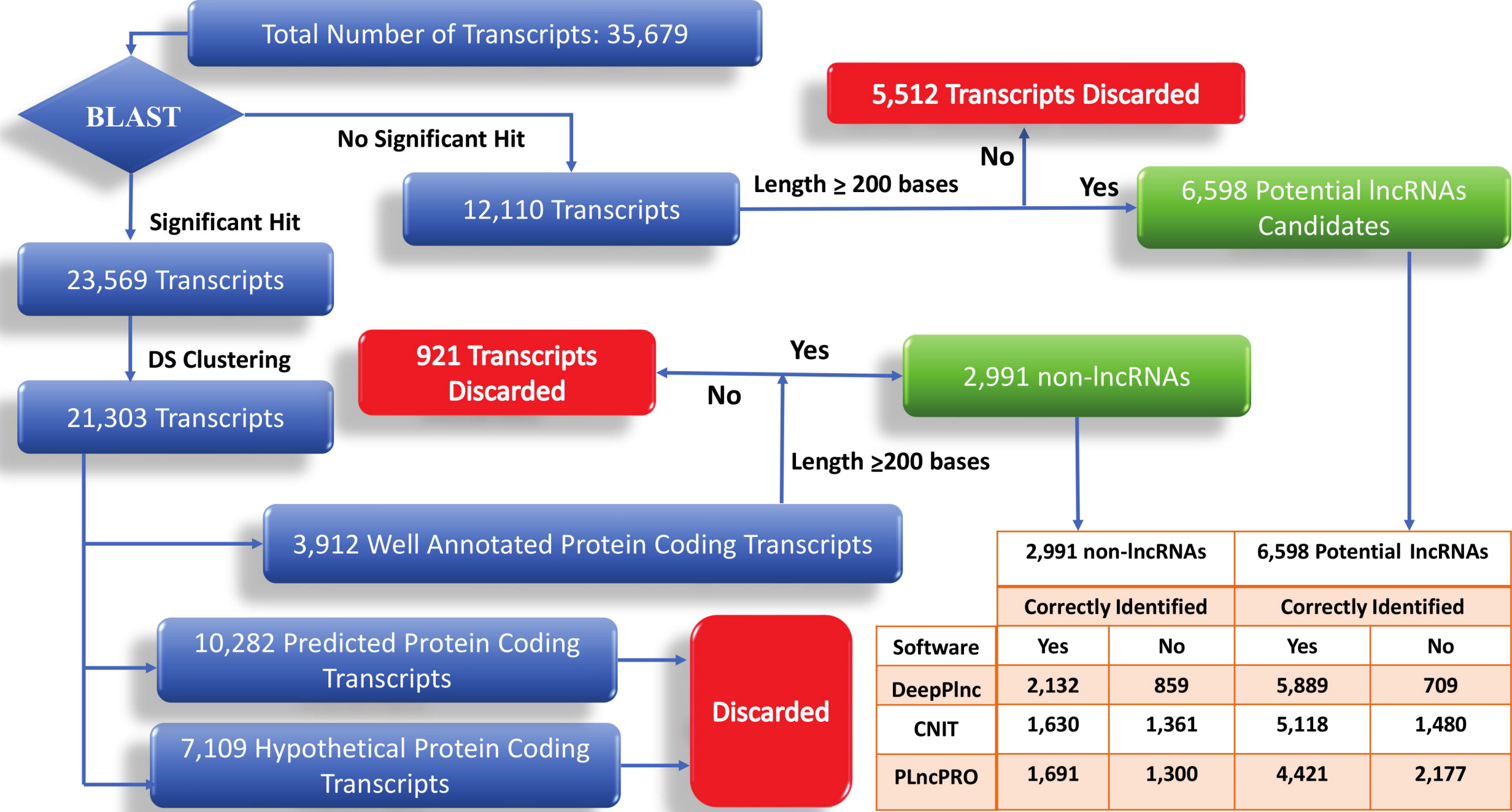

200 Sliding Window
400 Sliding Window
600 Sliding Window
800 Sliding Window
Accuracy (%)
92
90
88
86
84
82
80
78
76
MONOMER
STRUCTURE
MONOMER+STRUCTURE
DI+TRI
DI+TRI+MONO+STR
MONO+STR+H_bond+Pu-Py
Features

b)

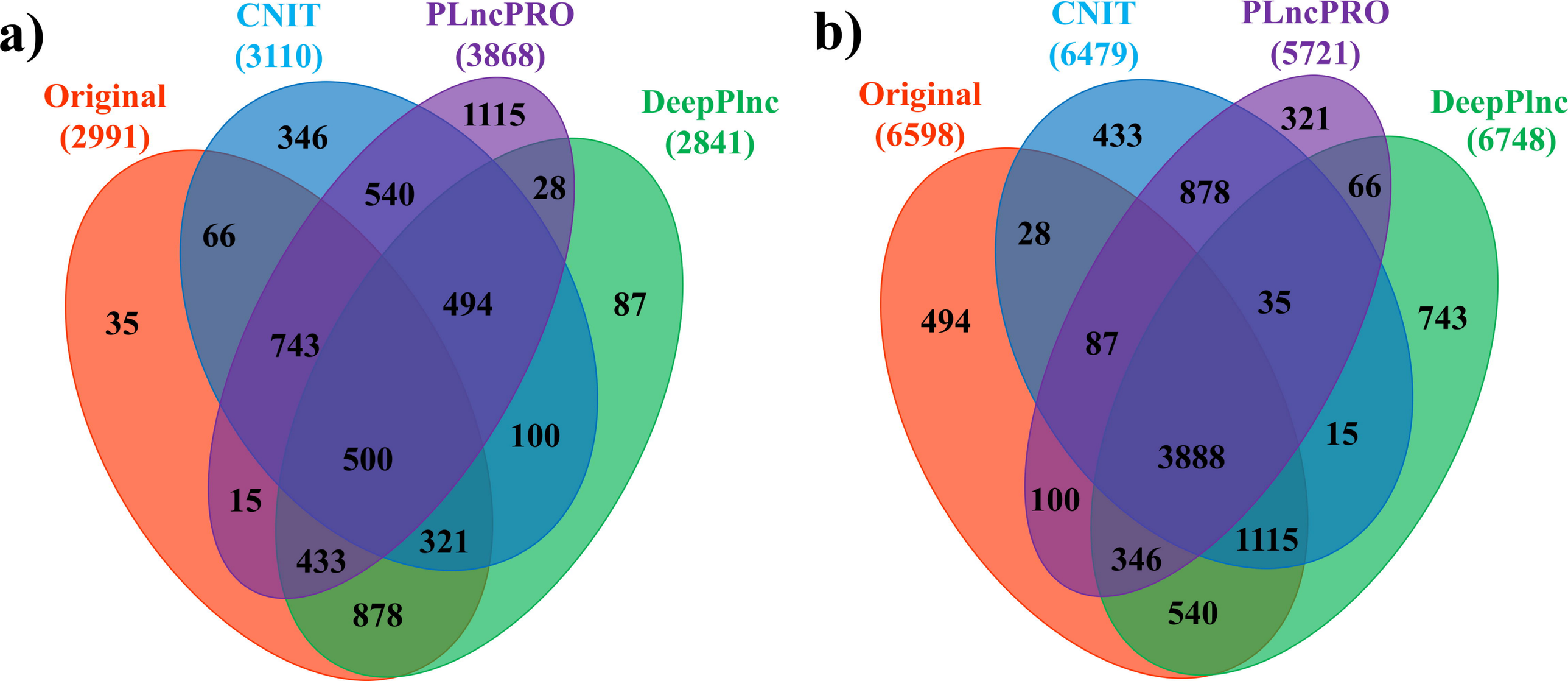

Full
Chunk
Accuracy (%)
95.0
92.5
90.0
87.5
85.0
82.5
80.0
77.5
75.0
1
2
3
4
5
6
7
8
9
10
Cross fold

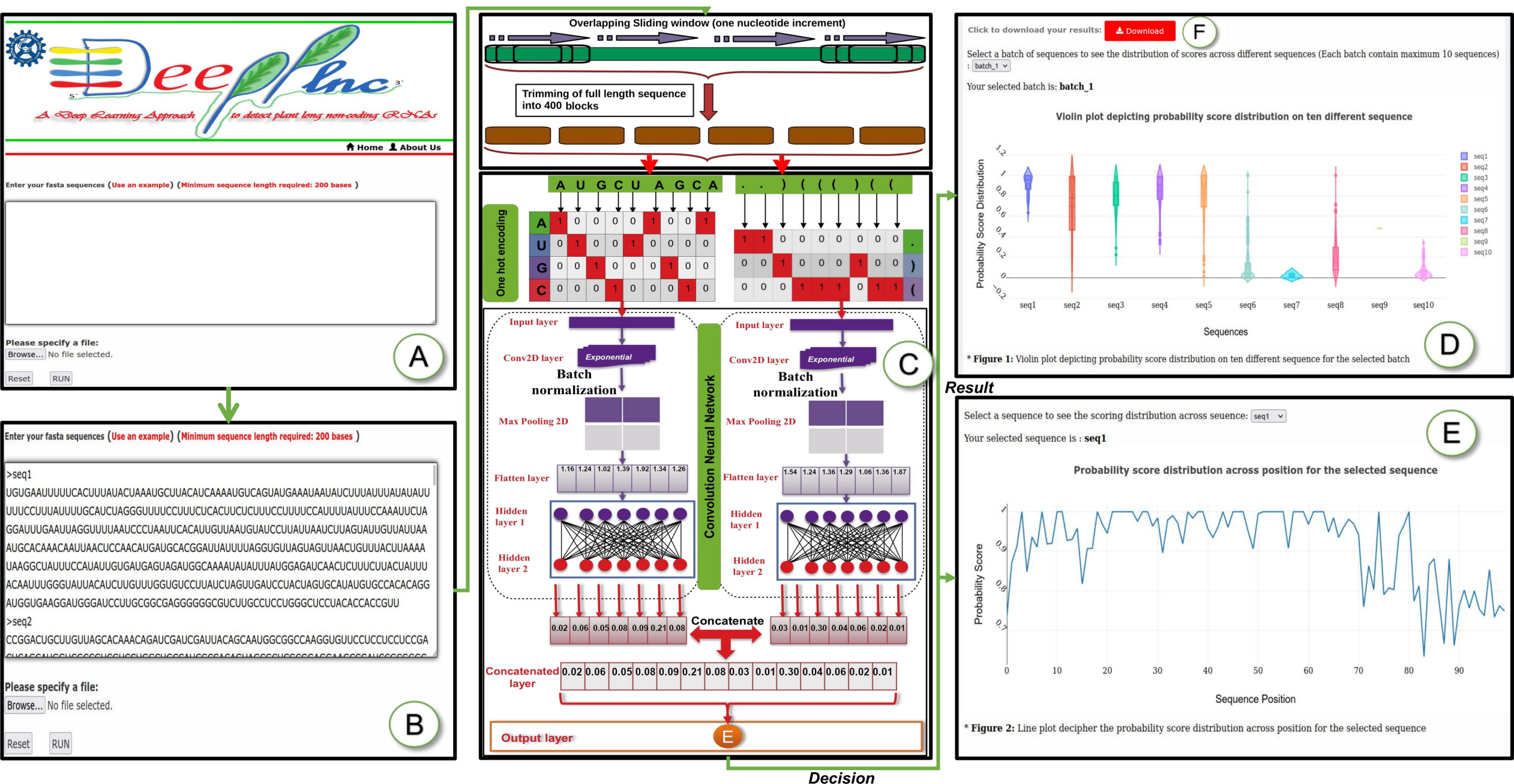

a)
Chunk Dataset
Tools
RNAplonc 61.35 23.30 100.00 37.79 36.19
PreLnc(A.th) 70.04 40.93 99.16 57.74 49.31
PreLnc(z.m.) 82.59 76.44 88.74 81.45 65.69
PLncPRO(d) 87.86 83.49 92.23 87.31 76.01
PLncPRO(m) 88.52 86.78 90.27 88.32 77.09
CNIT 87.07 84.46 89.64 86.62 74.22
DeepPlnc 92.01 87.95 96.06 91.25 84.28
PlncRNA-HDeep 65.66 40.93 90.38 54.38 36.02
Accuracy
SENSITIVITY
SPECIFICITY
F1-SCORE
MCC
Performance metrics
100
90
80
70
60
50
40
30
b)
Full Dataset
Tools
RNAplonc 92.20 89.67 95.13 92.50 84.58
PreLnc(A.th) 90.30 95.53 85.07 90.78 81.05
PreLnc(z.m.) 92.60 91.60 93.60 92.53 85.22
PLncPRO(d) 92.08 92.47 91.70 92.11 84.17
PLncPRO(m) 75.07 89.57 60.57 78.22 52.38
CNIT 87.95 84.09 91.80 87.45 76.12
DeepPlnc 96.16 96.46 95.86 95.46 90.25
PlncRNA-HDeep 74.62 88.03 61.20 77.62 51.11
Accuracy
SENSITIVITY
SPECIFICITY
F1-SCORE
MCC
Performance metrics
90
80
70
60
c)
Chunk Dataset
Tools
RNAplonc 56.25 12.61 99.98 22.39 25.87
PreLnc(A.th.) 66.58 34.42 98.75 50.74 43.32
PreLnc(z.m.) 72.19 65.44 78.93 73.70 44.78
PLncPRO(d) 87.53 76.36 98.70 85.96 77.01
PLncPRO(m) 87.85 77.76 97.94 86.49 77.29
CNIT 77.36 98.50 56.22 81.31 60.38
DeepPlnc 90.23 90.01 90.45 90.22 80.46
PlncRNA-HDeep 73.83 66.42 81.23 71.73 48.18
Accuracy
SENSITIVITY
SPECIFICITY
F1-SCORE
MCC
Performance metrics
90
80
70
60
50
40
30
20
d)
Full Dataset
Tools
RNAplonc 92.63 92.37 92.89 92.61 85.26
PreLnc(A.th.) 90.77 92.18 89.38 90.91 81.59
PreLnc(z.m.) 78.87 93.93 63.81 81.64 60.56
PLncPRO(d) 93.47 93.66 93.27 93.48 86.93
PLncPRO(m) 92.98 93.44 92.51 93.01 85.96
CNIT 86.82 87.16 86.47 86.86 73.63
DeepPlnc 96.02 97.48 95.31 96.24 92.04
PlncRNA-HDeep 79.07 91.56 66.58 81.39 60.04
Accuracy
SENSITIVITY
SPECIFICITY
F1-SCORE
MCC
Performance metrics
95
90
85
80
75
70
65


a)
Chunks Dataset
True Positive rate
False Positive Rate
PreLnc_ath_chunks = 97.67
PreLnc_zma_chunks = 89.99
PLncPRO_dicot_chunks = 95.08
PLncPRO_monocot_chunks = 99.10
PlncRNA-HDeep_chunks = 73.25
DeepPlnc_chunks = 99.78
Random
b)
Full Dataset
True Positive rate
False Positive Rate
PreLnc_ath_full = 97.00
PreLnc_zma_full = 88.04
PLncPRO_dicot_full = 96.94
PLncPRO_monocot_full = 95.36
PlncRNA-HDeep_full = 85.74
DeepPlnc_full = 99.24
Random
c)
Chunks Dataset
True Positive rate
False Positive Rate
PreLnc_ath_chunks = 89.45
PreLnc_zma_chunks = 90.56
PLncPRO_dicot_chunks = 92.08
PLncPRO_monocot_chunks = 91.92
PlncRNA-HDeep_chunks = 80.64
DeepPlnc_chunks = 94.01
Random
d)
Full Dataset
True Positive rate
False Positive Rate
PreLnc_ath_full = 97.00
PreLnc_zma_full = 92.13
PLncPRO_dicot_full = 96.65
PLncPRO_monocot_full = 98.70
PlncRNA-HDeep_full = 89.00
DeepPlnc_full = 99.21
Random



	2,991 non-lncRNAs		6,598 Potential lncRNAs	
	Correctly Identified		Correctly Identified	
Software	Yes	No	Yes	No
DeepPlnc	2,132	859	5,889	709
CNIT	1,630	1,361	5,118	1,480
PLncPRO	1,691	1,300	4,421	2,177


a)
Original
(2991)
CNIT
(3110)
PLncPRO
(3868)
DeepPlnc
(2841)
35
346
1115
66
540
28
494
87
743
100
500
15
321
433
878
b)
Original
(6598)
CNIT
(6479)
PLncPRO
(5721)
DeepPlnc
(6748)
494
433
321
28
878
66
35
743
87
15
3888
100
1115
346
540


DeepLnc
A Deep Learning Approach to detect plant long non-coding RNAs
Home
About Us
Enter your fasta sequences (Use an example) (Minimum sequence length required: 200 bases)
Please specify a file:
Browse... No file selected.
Reset
RUN
A
>seq1
UGUGAAUUUUUCACUUUAUACUAAAUGCUUACAUCAAAAUGUCAGUAUGAAAUAAUAUCUUUAUUUAUAUAUU
UUUCCUUUAUUUUGCAUCUAGGGUUUUCCUUUCUCACUUCUCUUUCCUUUUCCAUUUUAUUUCCAAAUUCUA
GGAUUUGAAUUAGGUUUUAAUCCCUAAUUCACAUUGUUAAUGUAUCCUUAUUAAUCUUAGUAUUGUUAUUAA
AUGCACAAACAAUUAACUCCAACAUGAUGCACGGAUUAUUUUAGGUGUUAGUAGUUAACUGUUUACUUAAAA
UAAGGCUAUUUCCAUAUUGUGAUGAGUAGAUGGCAAAAUAUAUUUAUGGAGAUCAACUCUUUCUUACUAUUU
ACAAUUUGGGUAUUACAUCUUGUUUGGUGUCCUUAUCUAGUUGAUCCUACUAGUGCAUAUGUGCCACACAGG
AUGGUGAAGGAUGGGAUCCUUGCGGCGAGGGGGGCGUCUUGCCUCCUGGGCUCCUACACCACCGUU
>seq2
CCGGACUGCUUGUUAGCACAAACAGAUCGAUCGAUUACAGCAAUGGCGGCCAAGGUGUUCCUCCUCCUCCGA
B
Overlapping Sliding window (one nucleotide increment)
Trimming of full length sequence into 400 blocks
One hot encoding
A U G C U A G C A
. .) ((( ) ((
Input layer
Conv2D layer
Exponential
Batch normalization
Max Pooling 2D
Flatten layer
Hidden layer 1
Hidden layer 2
Convolution Neural Network
C
Concatenate
Concatenated layer
Output layer
E
Decision
Click to download your results:
Download
F
Select a batch of sequences to see the distribution of scores across different sequences (Each batch contain maximum 10 sequences) : batch_1
Your selected batch is: batch_1
Violin plot depicting probability score distribution on ten different sequence
Probability Score Distribution
Sequences
seq1
seq2
seq3
seq4
seq5
seq6
seq7
seq8
seq9
seq10
D
* Figure 1: Violin plot depicting probability score distribution on ten different sequence for the selected batch
Result
Select a sequence to see the scoring distribution across seuence: seq1
Your selected sequence is : seq1
Probability score distribution across position for the selected sequence
Probability Score
Sequence Position
E
* Figure 2: Line plot decipher the probability score distribution across position for the selected sequence