Long non-coding RNA functional annotation : machine learning approaches

Zhang, Yu

2021

Zhang, Y. (2021). Long non-coding RNA functional annotation : machine learning approaches. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/154930>

<https://hdl.handle.net/10356/154930>

<https://doi.org/10.32657/10356/154930>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 20:08:12 SGT

LONG NON-CODING RNA FUNCTIONAL
ANNOTATION: MACHINE LEARNING
APPROACHES

ZHANG YU

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

03/06/2021

.....
Date

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
ZHANG Yu 张 宇
.....
[Input Name Here & Sign above]

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

26/07/2021

Date

Kwoh Chee Keong

[Input Supervisor Name Here & Sign above]

Authorship Attribution Statement

This thesis contains material from four papers published in the following peer-reviewed journals in which I am listed as an author.

Chapter 3 is published as Y. Zhang⁺, Y. Cai⁺, X. Roca, M. J. Fullwood, and C. K. Kwoh, “Chromatin loop anchors predict transcript and exon usage”, *Briefings in Bioinformatics*, 2021. DOI: <https://doi.org/10.1093/bib/bbab254>.

The contributions of the co-authors are as follows:

- I conceived of the research, designed, performed, and analyzed the experiments of verified transcript prediction and exon usage prediction, and prepared the manuscript drafts.
- Dr. Cai conceived of the research, prepared the data, designed, performed, and analyzed the experiments of transcription abundance prediction, and prepared the manuscript drafts.
- A/Prof Roca advised on the background and discussion of alternative splicing and exon usage and edited the manuscript drafts.
- A/Prof Fullwood provided the initial project direction, conceived of the research, and edited the manuscript drafts.
- A/Prof Kwoh advised on the machine learning analyses and edited the manuscript drafts.

Chapter 4 is published as Y. Zhang, C. Jia, M. J. Fullwood, and C. K. Kwoh, “DeepCPP: a deep neural network based on nucleotide bias information and minimum distribution similarity feature selection for RNA coding potential prediction”, *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 2073–2084, 2021. DOI: <https://doi.org/10.1093/bib/bbaa039>.

The contributions of the co-authors are as follows:

- I designed the study, performed all the experiment, analyzed the results, and prepared the manuscript drafts.
- A/Prof Jia provided the initial project direction, co-designed the study, assisted in the data collection, and edited the manuscript drafts.

- A/Prof Fullwood and A/Prof Kwoh advised on the study frameworks and edited the manuscript drafts.

Chapter 5 is published as Y. Zhang, C. Jia, and C. K. Kwoh, "Predicting the interaction biomolecule types for lncRNA: an ensemble deep learning approach", *Briefings in Bioinformatics*, 2020. DOI: <https://doi.org/10.1093/bib/bbaa228>.

The contributions of the co-authors are as follows:

- I provided the initial project direction, designed the study, performed all the experiment, analyzed the results, and prepared the manuscript drafts.
- A/Prof Jia and A/Prof Kwoh advised on the study frameworks and edited the manuscript drafts.

Chapter 6 is published as Y. Zhang, Y. Long, and C. K. Kwoh, "Deep learning based DNA: RNA triplex forming potential prediction," *BMC Bioinformatics*, vol. 21, p. 522, 2020. DOI: <https://doi.org/10.1186/s12859-020-03864-0>.

The contributions of the co-authors are as follows:

- I provided the initial project direction, designed the study, performed all the experiment, analyzed the results, and prepared the manuscript drafts.
- Dr. Long and A/Prof Kwoh advised on the study frameworks and edited the manuscript drafts.

20/07/2021

.....
Date

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
ZHANG Yu
.....
[Input Name Here & Sign Above]

Acknowledgments

First, I want to say thanks to my esteemed supervisor - Prof. Kwoh Chee Keong for his supervision and support for my PhD degree. Prof. Kwoh gave me the chance to become a PhD student and start my research career. He also gave me the opportunity to know lots of researchers and work with them. He is such a nice person that always encourages me to try what I thought and respects my decisions. His kind and academical supports give me the confidence to finish my PhD research.

Next, I with to thank Prof. Melissa Jane Fullwood. She provides me the valuable opportunities to collaborate with her lab, where I learnt a lot. And also, I want to express my gratefulness to Dr. Jia Cangzhi for helping me in my first year of PhD. Without Dr. Jia Cangzhi's help, my research would not be so smooth. Besides, my sincere thanks also go to Dr. Yin Rui, Dr. Zhou Xinrui, Mr. Lin Zhuoyi, Mr. Long Yahui, and Dr. Cai Yichao, who have provided valuable suggestions and support to my researches. Furthermore, I also wish to thank all the people from Biomedical Informatics Lab for providing a warm and friendly environment, and thank the School of Computer Science and Engineering and Nanyang Technological University for providing an excellent research environment and resources.

Finally yet importantly, I am extremely grateful to my parents. They unselfishly support me throughout my PhD study. Their love, care, and encouragement help me to go through the tough time and finish my PhD study.

Abstract

Long Non-coding RNA (lncRNA) is an enormous class of RNA molecules with transcript length longer than 200 nucleotides. lncRNAs play dominant roles in complex pathological and physiological processes, and they are implicated to be related to diseases. Although functional studies suggest significant roles for lncRNAs, only a few of them are well characterized. lncRNA functional annotation mainly includes two parts: 1) lncRNA annotation, and 2) lncRNA function exploration. The biological experiments for lncRNA functional annotation are costly and time-intensive. Moreover, the characteristics of lncRNAs such as the high biogenesis diversity, low conservation levels, and low expression levels pose further challenges to the understanding of lncRNAs. The goal of this thesis is to explore the lncRNA functional annotation using machine learning approaches.

First, we aim to predict transcript and exon usage using a machine learning approach. Whether the chromatin interaction-related data could contribute to the transcript and exon usage prediction is unclear yet. We use a diverse range of transcription factors and Chromatin Interaction Analysis with Paired-end Tags (ChIA-PET) data which indicates locations of chromatin loops in the genome from different cell lines. We demonstrate that both transcript and exon usage can be effectively predicted via different validation methods. We find that RNA Polymerase II ChIA-PET data is one of the most important features in both transcript and exon usage prediction, suggesting that chromatin loop anchors are predictive of transcription and exon usage.

Second, we manage to distinguish lncRNA from messenger RNA (mRNA) after the transcript annotation. We represent the information around the start codon into a single feature named nucleotide bias, which is shown to work effectively in this classification problem. We then develop DeepCPP for *human*, vertebrate, and insect species to distinguish between lncRNA and mRNA. We demonstrate that DeepCPP performs better

than other computational methods, especially on the transcript with a small open reading frame, which used to be a bottleneck.

Third, we move to the lncRNA function prediction after the lncRNA annotation. We investigate the functions of lncRNAs from a new perspective by indicating the types of biomolecules that are likely to interact with a lncRNA. To achieve this, we explore the main mechanisms in the interactions of lncRNA with other biomolecules and we develop a deep learning framework by ensembling 5 models. We show that it is possible to predict types of interaction biomolecules for a lncRNA. According to our collected data and the prediction results from our model, we speculate that the attributes of lncRNAs interacting with DNA might be dissimilar to those interacting with merely RNA.

Finally, we focus on DNA:lncRNA triplex formation, a certain type of interaction to reveal the lncRNA function. We present a deep learning tool to identify the most possible lncRNAs and DNA sites in forming triplex in practice. As far as we know, this is the first machine learning model in DNA:lncRNA triplex prediction. The high performances of our model indicate that we overcome some limitations of existing methods and show that the potential lncRNA and DNA sites to form triplex in practice are predictable.

Therefore, the analysis conducted in this thesis demonstrates the effectiveness of using machine learning approaches for lncRNA functional annotation.

Contents

Acknowledgments	v
Abstract	vi
List of Figures	xiii
List of Tables	xxvi
List of Abbreviations	xxviii
List of Publications	xxxiii
1 Introduction	1
1.1 Background	1
1.1.1 Transcript, Exon, and Coding RNA	1
1.1.2 Long Non-coding RNA (lncRNA)	1
1.1.3 The Role and Application of lncRNA	2
1.2 Motivation and Objective	3
1.3 Major Contributions	4
1.4 Thesis Organization	6
2 Literature Review	7
2.1 Experimental Methods	7
2.2 Databases	8
2.2.1 National Center for Biotechnology Information (NCBI)	8
2.2.2 University of California Santa Cruz Genome Browser website (UCSC)	9
2.2.3 Ensembl	9
2.3 Computation Methods	10
2.3.1 RNA Transcript and Exon Usage Prediction	10

2.3.2	lncRNA and mRNA Classification	11
2.3.3	lncRNA Function Prediction	13
2.4	Evaluation Matrix	15
2.5	Chapter Summary	18
3	Chromatin Loop Anchors Predict Transcript and Exon Usage	19
3.1	Introduction	19
3.2	Materials and Methods	24
3.2.1	Data Collection and Definition	24
3.2.1.1	Transcript Prediction Feature Matrices	24
3.2.1.2	Exon Usage Prediction Feature Matrices	26
3.2.2	Model Construction	29
3.2.3	Validation Methods	29
3.2.4	Evaluation Criteria	30
3.3	Results	31
3.3.1	Verified Transcripts can be Predicted from Transcription Factors and ChIA-PET Data	31
3.3.2	Transcription Abundance Prediction	37
3.3.3	Exon Usage Coefficient Prediction	38
3.4	Discussion	42
3.5	Chapter Summary	50
4	RNA Coding Potential Prediction with Nucleotide Bias Information	53
4.1	Introduction	53
4.2	Materials and Methods	55
4.2.1	Dataset	55
4.2.2	Feature Extraction	58
4.2.2.1	The Maximum ORF Length and Coverage	60
4.2.2.2	Hexamer Score	60
4.2.2.3	Fickett Score	60
4.2.2.4	k-mer, g-gap, and g-bigap	61
4.2.2.5	Nucleotide Bias Feature	62

4.2.3	Model Construction	63
4.2.4	Evaluation Criteria	63
4.3	Results and Discussion	65
4.3.1	Feature Exploration	65
4.3.2	Feature Selection	66
4.3.3	Performance of DeepCPP <i>human</i> Model	70
4.3.4	Performance of DeepCPP Integrate Models	72
4.4	Chapter Summary	77
5	Predicting the Types of Biomolecules Interacting with lncRNAs: An Ensemble Deep Learning Approach	79
5.1	Introduction	79
5.2	Materials and Methods	83
5.2.1	Data Description	83
5.2.2	Feature Extraction	85
5.2.3	Model Construction	87
5.2.4	Evaluation Criteria	89
5.3	Results and Discussion	89
5.3.1	Interaction Mechanisms between lncRNA and Biomolecules	89
5.3.2	Performances of lncIBTP	89
5.3.3	lncIBTP as a Pre-step for Specific lncRNA-molecule Prediction	94
5.3.4	A Case Study	95
5.3.5	Cross-species Validation	96
5.4	Chapter Summary	96
6	Deep Learning-based DNA:lncRNA Triplexes Prediction	99
6.1	Introduction	99
6.2	Materials and Methods	101
6.2.1	Dataset	101
6.2.1.1	Triplex lncRNA Dataset	101
6.2.1.2	Triplex DNA Site Dataset	103
6.2.2	Feature Extraction	104

6.2.3	Model Construction	104
6.2.4	Evaluation Criteria	105
6.3	Results and Discussion	106
6.3.1	Feature and Data Exploration	106
6.3.2	Performance of Triplex lncRNA Prediction Model	107
6.3.3	Performance of Triplex DNA Site Prediction Model	109
6.3.4	TriplexFPP Model Interpretation	111
6.3.5	Limitation	111
6.3.6	An Exploration on <i>in cis</i> / <i>in trans</i> Triplex lncRNA	115
6.4	Chapter Summary	117
7	Conclusions and Future Directions	118
7.1	Conclusions	118
7.2	Future Directions	120
7.2.1	Improve upon Current Approaches	121
7.2.2	New Application Opportunities	121
7.2.3	lncRNA Identification	121
7.2.4	lncRNA Function Prediction	122
	Appendices	126
A	Model Determination and Parameter Tuning in Chapter 3	126
A.1	Parameter Determination for transcription prediction	126
B	Model Determination and Parameter Tuning in Chapter 4	129
B.1	Comparisons among Different Machine Learning Models	129
B.2	Network Parameter Determination	131
C	Model Determination and Parameter Tuning in Chapter 5	134
C.1	Network Parameter Determination	134
C.2	Determination of Ensemble	134
C.3	Parameter Determination for Other Models	136

D Model Determination and Parameter Tuning in Chapter 6	138
D.1 Network Parameter Determination	138
D.2 Parameter Determination for Other Models	141
References	144

List of Figures

1.1	The demonstration of the relationship among RNA transcript, exon, mRNA, and lncRNA.	2
1.2	The subtypes of lncRNAs grouped by the genome location to the nearest protein-coding genes. Sense lncRNAs and antisense lncRNAs cross exons of genes that can encode to proteins on the coding strands and complementary strands separately, intronic lncRNAs locate within the intronic regions of genes that can encode to proteins, long intergenic ncRNAs (lincRNAs) locate between 2 genes that encode to proteins, and bidirectional lncRNAs transcribed opposite to the transcriptional initial locus of protein-coding genes.	3
1.3	The problem formulation in lncRNA function annotation and the objectives in this thesis.	5
3.1	The workflow to conduct the study.	23
3.2	The prediction results in verified transcript prediction. Sn (sensitivity) represents the positive prediction accuracy and Sp (specificity) represents the negative prediction accuracy. a. 10-fold validation. The boxplot here shows the spread and center of the 10-fold validation results. The five horizon lines from bottom to top represent the minimum (the smallest number in the data set), the first to the third quartile, and the maximum (the largest number in the data set), respectively. b. Chromosome-split validation. c. Cross-cell validation.	33
3.3	Feature importance in verified transcript prediction.	34

3.4	The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. Acc: Accuracy, AUROC: Area Under Receiver Operating Characteristic curve, AUPRC: Area Under Precision-Recall Curve, Sn: Sensitivity, Sp: specificity.	34
3.5	The performance comparison among baseline models with the average values from the 10-fold validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. GB: Gradient Boosting Trees, Adaboost: Adaptive Boosting (n_estimators=100), RF: Random Forest (n_estimators=100), CNN: Convolutional Neural Network (Conv1D(filters=40, kernel_size=2, strides=1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512), NN: Deep Neural Network (dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512).	35
3.6	The performance comparison among baseline models from the chromosome-split validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: n_estimators=100, RF: n_estimators=100, CNN: Conv1D(40, 2, 1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512.	36
3.7	Histogram plot for TPM values of three cell lines. From top to bottom: K562, HeLaS3, and GM12878. Left: TPM values $\leq Q3$, right: TPM values $> Q3$	39

3.8	Results for transcription abundance prediction. a. The prediction results of 10-fold validation. The bottom table represents average accuracy. b. The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation for GM12878, HeLaS3, and K562 cell. Acc: Accuracy, AUROC: Area Under Receiver Operating Characteristic curve, AUPRC: Area Under Precision-Recall Curve, Sn: Sensitivity, Sp: specificity. c. The performance comparison among baseline models with the average values from the 10-fold validation for GM12878, HeLaS3, and K562 cell. AdaBoost: n_estimators=100, RF: n_estimators=100, CNN: Conv1D(40, 2, 1, activation ='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512. d. The performance comparison among baseline models (parameters are same as above) from the chromosome-split validation for GM12878, HeLaS3, and K562 cell.	40
3.9	Feature importance in transcript abundance prediction.	41
3.10	The prediction results in exon usage prediction. Sn (sensitivity) represents the positive prediction accuracy and Sp (specificity) represents the negative prediction accuracy. a. 10-fold validation. The boxplot here shows the spread and center of the 10-fold validation results. The bottom table represents the random values. b. Chromosome-split validation. The black horizon lines represent the random results. c. Cross-cell validation. The black horizon lines represent the random results.	43
3.11	Feature importance in exon usage prediction.	44
3.12	The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell.	44

3.13	Example of exon that has predicted exon usage in GM12878 but not HeLaS3 cell. RNA-Seq signal of H1hESC, K562, GM12878, and HeLaS3 cells are shown in RPKM. Example exons are highlighted in light blue. The example exons in this figure are those predicted as positive in GM12878 while negative in HeLaS3 cell using K562 model. For each example, a total RNA-Seq signal across the exon region is calculated and plotted in barplot. We can see that in this category, the total RNA-Seq signal of GM12878 is higher than HeLaS3. The examples showed here are: a. ENSE00003890465.1; b. ENSE00000767123.1; c. ENSE0000383698.1. . .	45
3.14	Example of exon that has predicted exon usage in HeLaS3 but not GM12878 cell. RNA-Seq signal of H1hESC, K562, GM12878, and HeLaS3 cells are shown in RPKM. Example exons are highlighted in light blue. The example exons in this figure are those predicted as positive in HeLaS3 while negative in GM12878 cell using K562 model. For each example, a total RNA-Seq signal across the exon region is calculated and plotted in barplot. We can see that in this category, the total RNA-Seq signal of HeLaS3 is higher than GM12878. The examples showed here are: a. ENSE00001904186.1; b. ENSE00003508391.1 and ENSE00000735744.1; c. ENSE00000801685.4.	46
3.15	The performance comparison among baseline models with the average values from the 10-fold validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: n_estimators=100, RF: n_estimators=100, CNN: Conv1D(40, 2, 1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512.	47

3.16	The performance comparison among baseline models from chromosome-split validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: <code>n_estimators=100</code> , RF: <code>n_estimators=100</code> , CNN: Conv1D(40, 2, 1, activation = 'relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512.	48
3.17	Schematic of our proposed model of alternative splicing, which gives rise to differential exon usage, chromatin loops and transcription machinery. In this model, we speculate that HMs such as H3K36me3, H3K9ac, and chromatin interactions, can bind to splicing regulators and transcription machinery. This would explain our findings that these marks predict high exon usage.	51
4.1	The description of data amount, dataset annotation, and the procedure to build datasets. The number in each text box is the data amount left after filtering in the last step. The color of the text box is to distinguish ncRNA and mRNA, vertebrate and insect.	58
4.2	The workflow of DeepCPP. a. Overall structure. b. Feature extraction approaches and the sizes of each feature subset. c. Convolutional Neural Network structure, including two CNN layers, a flatten layer and a dropout layer with dropout rate of 20%, and two dense layers.	64
4.3	Evaluation of new features used in this work. a-c. Sequence logo representation of the training dataset for a. <i>human</i> species, b. vertebrate species, and c. insect species. The first row and the second row are corresponding to lncRNA and mRNA separately. d-e. Performance of each feature set on d. CPPred <i>human</i> test dataset D_{test_H} and e. CPPred <i>human</i> sORF test-dataset $D_{test_H_sORF}$. f-g. ROC (left) and PRC curve (right) for continuous or discontinuous k-mer feature sets on f. D_{test_H} and g. $D_{test_H_sORF}$	67
4.4	Accuracies achieved by different sizes of feature subsets. a. normal type data D_{test_H} , and b. sORF type data $D_{test_H_sORF}$	69

4.5	McNemar test applied on DeepCPP and other methods on 4 previous test datasets. Each table tabulates the correctly or wrongly predicted samples by each model, the McNemar test statistic outcomes and the datasets with data amounts for each class are demonstrated below the table. The filtered datasets in the second row only contain RNAs whose lengths > 200. . . .	72
4.6	Evaluation of DeepCPP integrated models. a-d. Comparisons among DeepCPP, CPPred, and Hugo's SVM method on vertebrate and insect datasets a. D_{test_V} , b. $D_{test_V_sORF}$, c. D_{test_I} , and d. $D_{test_I_sORF}$. e. Comparisons between DeepCPP and CPPred on specific test datasets. From left to right: D_{test_M} , $D_{test_M_sORF}$, normal type data from species of <i>fruit fly</i> and <i>mosquito</i> , normal type data from species of <i>honeybee</i> , sORF type data from species of <i>fruit fly</i> and <i>mosquito</i> , and sORF type data from species of <i>honeybee</i> . CPPred - Integrated model and hugo's SVM - GRCm38_GRCz10 model are adopted for the comparisons.	78
5.1	Some main interaction mechanisms in the interactions of lncRNA-RNA. a. lncRNA serves as the hosts to some small ncRNAs. b. lncRNA binds to RNA in a direct way. c. lncRNA involved in the ceRNA network to regulate mRNA through miRNA. d. lncRNA competes with miRNA on mRNA binding sites.	84
5.2	The main interaction mechanisms in interactions of lncRNA-DNA. a. lncRNA binds to DNA through the DNA-binding protein. b. lncRNA forms R-loop with DNA. c. lncRNA forms triplex with DNA.	84

5.3	Construction of Dataset. a. The Venn diagram of lncRNA in two databases according to their reported interacted biomolecules. The numbers of lncRNA genes in each type of interaction for each database are shown. lncRNAs interacting with multiple types of biomolecules are shown as the overlap. Left: NPInter v4.0 database. 5 types of lncRNAs are found in this database based on their interactions with i) only RNA, ii) only protein, iii) RNA and protein, iv) DNA and protein, and v) DNA, RNA, and protein. Right: lncRInter database. Besides the same 5 types of lncRNAs found in NPInter database, lncRInter database has one more type of lncRNA: interacting with only DNA. b. The constructed dataset which integrating the information from lncRInter and NPInter v4.0 databases. The lncRNAs are finally divided into four classes according to their reported interactions with i) only RNA, ii) only protein, iii) RNA and protein, and iv) interacting with DNA-related (only DNA, DNA and protein, and DNA, RNA, and protein). The numbers of lncRNA samples in each class are shown as annotated by GENCODE release 33 / after removing redundancy with a cut-off of 0.8. c. The <i>mouse</i> dataset. The lncRNAs are collected from NPInter 4.0 and the numbers of lncRNA samples in each class are shown after contrasting with GENCODE m24 annotation.	86
5.4	The performances of different feature sets evaluated by the model trained with one CNN layer and fifty epochs.	87
5.5	The construction of lncIBTP. Each model includes 4 CNN layers, 1 flatten layer, 1 dropout layer with dropout rate of 20%, 1 dense layer with activation function of ReLU, another 1 dropout layer with dropout rate of 20%, and another 1 dense layer with activation function of ReLU. Class 0 to 3 are corresponding to lncRNA samples that interacting with only RNA, only protein, RNA and protein, and DNA-related, respectively.	88

5.6	A summary of the interaction mechanisms between lncRNAs and other biomolecules based on <i>human</i> data from NPInter v4.0 database. a. The ratio of RNA types to interact with lncRNAs, where the miRNA and mRNA take up the majority parts. b. The number of the cases for different interaction modes in different biomolecules associated with lncRNAs, representing in the form of log2. c. Some lncRNAs are shown to associate with multiple molecules whereas some only are reported to associate with a single molecule. Color-molecule: purple-lncRNA, red-RNA, green-protein, blue-DNA. The black straight line denotes the interaction.	90
5.7	The performances of lncIBTP. a. The comparison between lncIBTP and other baseline methods via the 5-fold cross-validation represented by the boxplots. b. The confusion matrix using each fold as test data in 5-fold cross-validation. x-axis and y-axis denote the predicted label and the true label separately. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively. . .	92
5.8	The illustration of the ROC (left) and PRC (right) curves for each class in the 5-fold cross-validation.	93
5.9	Confusion matrix of RNAInter prediction. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively.	95
5.10	The reported and the predicted types of biomolecules to be interact with SOCAR, RAIN, LINC02407, and LINC01980.	95
5.11	The confusion matrix for <i>mouse</i> data. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively.	97
5.12	Feature heatmaps for 4 classes in <i>human</i> and <i>mouse</i> dataset. Top: <i>human</i> data, bottom: <i>mouse</i> data. The heatmap from left to right is corresponding to class of RNA, protein, RNA and protein, and DNA-related, respectively. The original features (ordered as 3-mer, 2-bigap, and 3-bigap with nucleotide order of <i>ACGT</i>) are reshaped to size of 32×18	97
6.1	A brief illustration of data collection.	103

6.2	The construction of TriplexFPP. TriplexFPP contains two individual models that corresponding to triplex lncRNA and DNA sites prediction. The detailed structures and parameters are illustrated in the figure.	105
6.3	Determining the features for TriplexFPP triplex lncRNA (left) and DNA sites (right) prediction model.	106
6.4	The nucleotide frequency distribution in positive data from different sources and negative data in lncRNA triplex prediction datasets.	107
6.5	The demonstration of the number of variants for each gene, the light color to dark color means a small variant number to a large variant number, outer: triplexlncRNA, inner: reported triplex lncRNA.	109
6.6	Prediction results of TriplexFPP triplex lncRNA model. a. The comparisons between CNN and other baseline methods (SVM, RF, Gradient Boosting, and NN) via the 5-fold cross-validation represented by boxplots. b. The comparisons among original dataset and datasets with removed redundancy ($th = 0.9$ and $th = 0.8$) via the 5-fold cross-validation. c. Radar chart for leave-out validation.	110
6.7	Prediction results of TriplexFPP triplex DNA sites model. a. The comparisons between CNN and other baseline methods (SVM, RF, Gradient Boosting, and NN) via for the 5-fold cross-validation represented by the boxplots. b. The predicted probability score distribution for validation data from the first fold.	111
6.8	The distribution of the probability scores predicted by TriplexFPP for the other 4 folds of cross-fold validation in triplex DNA sites dataset.	112

6.9	An interpretation of the TriplexFPP triplex lncRNA prediction model from the perspective of input features, where the kmerscore featuers are more distinguishable between positive and negative samples. The first row represents the average values of the original features which are ordered as kmerscore feature with k from 1 to 6, followed by k-mer features with k from 1 to 3 with the nucleotide order <i>ACGT</i> , and then the features were reshaped to a 9*10 matrix; the second and the third row represent the average values of the features obtained by 1 CNN layer and 2 CNN layers separately, where the x-axis and y-axis denote the filter and the 1st to the 15th convolution values, respectively. Left: positive training set, right: negative training set.	113
6.10	An interpretation of the TriplexFPP triplex DNA sites prediction model from the perspective of input features, where the kmerscore featuers are more distinguishable between positive and negative samples. The first row represents the average values of the original features which are ordered as kmerscore feature with k from 1 to 6, followed by k-mer features with k from 1 to 3 with the nucleotide order <i>ACGT</i> , and then the features were reshaped to a 9*10 matrix; the second and the third row represent the average values of the features obtained by 1 CNN layer and 2 CNN layers separately, where the x-axis and y-axis denote the filter and the 1st to the 15th convolution values, respectively. Left: positive training set, right: negative training set.	114
6.11	Statistic of <i>in cis</i> & <i>in trans</i> lncRNAs in TriplexlncRNA evaluated by TDF. Left: the number of lncRNAs in 2 types of targeting; right: the amounts of <i>in cis</i> binding sites for <i>in cis</i> & <i>in trans</i> targeting lncRNAs. .	116
6.12	Statistic of <i>in cis</i> & <i>in trans</i> targeting lncRNAs in TriplexlncRNA evaluated by TDF. Left: gene numbers in each type of targeting, right: variant numbers of PVT1 and LINC00963 in two types of targeting.	116
6.13	Gene names and number of variants for lncRNAs in two types of targeting.	116

7.1	An illustration of major groups of RNA. RNAs mainly can be divided into non-coding RNAs (ncRNAs) and messenger RNAs (mRNAs). ncRNAs consist of linear ncRNAs and circular RNAs (circRNAs). Linear ncRNAs consist of housekeeping ncRNAs and regulatory ncRNAs. Housekeeping ncRNAs consist of transfer RNAs (tRNAs), ribosomal RNAs (rRNAs), small nucleolar RNAs (snoRNAs), and small nuclear RNAs (snRNAs). Regulatory ncRNAs comprise of small ncRNAs and long non-coding RNAs (lncRNAs). Small ncRNAs contain PIWI-interacting RNAs (piRNAs), microRNAs (miRNAs), and small interfering RNAs (siRNAs).	123
A.1	The candidate and determined parameters for GB model in different transcription prediction task. The parameters were searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds.	127
A.2	The candidate and determined parameters for GB model in different exon usage prediction task. The parameters were searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds.. . . .	128
B.1	The comparison among different machine learning models using training data with 10-fold validation.	130
B.2	The candidate and determined parameters for SVM, RF, and GB model. The parameters were searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds.	130
B.3	DeepCPP normal prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions in dense layer, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. dense size of dense layer, and h. training epochs (the best test accuracy is marked by dash line).	132

B.4	DeepCPP sORF prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions in dense layer, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. dense size of dense layer, and h. training epochs (the best test accuracy is marked by dash line).	133
C.1	Network parameter determination. Accuracies achieved with different choices of a. sample numbers in each class (The models are trained with 1 convolutional layer (filter number=40, kernel size=2, strides=1), 50 epochs, and 20% dropout), b. numbers of convolution layers, c. dropout rates, d. activation functions, e. kernel size in the 1st convolution layer, f. optimizers, g dense layer sizes, and h. training epochs.	135
C.2	The performances with different number of ensemble models.	136
C.3	The candidate and determined parameters for baseline models. The parameters were searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds.	137
D.1	TriplexFPP triplex lncRNA prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. optimizers, and h. training epochs (the best test accuracy is marked by dash line).	139
D.2	TriplexFPP triplex DNA site prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. optimizers, and h. training epochs (the best test accuracy is marked by dash line).	140

D.3	The candidate and determined parameters for baseline models in predicting triplex lncRNA. The parameters for SVM, RF, and GB models are searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds and ‘recall’ scoring method.	142
D.4	The candidate and determined parameters for baseline models in predicting triplex DNA sites. The parameters for SVM, RF, and GB models are searched by <i>sklearn.model_selection.GridSearchCV</i> function in python with 5 folds and ‘recall’ scoring method.	143

List of Tables

2.1	The three genome browsers that will be used in this thesis.	10
2.2	The approaches in alternative splicing prediction discussed in this section.	12
2.3	The computational methods in lncRNA and mRNA classification discussed in this section.	14
2.4	The lncRNA function prediction approaches discussed in this section. . .	16
3.1	A summary of the cell lines used in this work.	25
3.2	TPM value to separate transcribed level. Q1, Q2, and Q3 represent the first, the second (median), and the third quartile in sorted TPM values, respectively.	26
3.3	The data amount in transcription prediction.	27
3.4	The data amount in exon usage prediction.	27
3.5	The list of features in each cell line.	28
3.6	Chromatin in training and test dataset when implementing the chromosome- split validation.	30
4.1	A summary of the datasets used in this work.	59
4.2	The choices of bin number in mDS feature ranking method and the sizes of the corresponding optimal feature subset on <i>human</i> normal type and sORF type test dataset.	69
4.3	Comparison between DeepCPP and other approaches on test datasets from the work of CPPred. The number of samples for the filtered test datasets are shown.	73

4.4	Summary of seven newly published lncRNA and mRNA distinction programs.	74
4.5	Comparison between DeepCPP <i>human</i> model and other approaches on newly constructed <i>human</i> test dataset.	75
4.6	Comparison between DeepCPP <i>human</i> model and other approaches on newly constructed <i>human</i> sORF type test dataset.	75
4.7	The number of correctly predicted mRNA samples with ORF lengths less than 100 codons by different approaches.	76
5.1	The AUROC (Area Under the ROC curves) and AUPRC (Area Under the PRC curves) values for each class in the 5 folds from cross-validation. . .	94
6.1	The binding information for reported triplex lncRNA.	115

List of Abbreviations

Acc	Accuracy
AUC	Area Under Curve
CDF	Cumulative Distribution Function
CDS	coding sequence
ceRNA	competing endogenous RNA
CHART-Seq	Capture Hybridization Analysis of RNA Targets Sequencing
ChIA-PET	Chromatin Interaction Analysis with Paired-end Tags
ChIP-Seq	Chromatin Immunoprecipitation Sequencing
ChIRP-Seq	Chromatin Isolation by RNA Purification Sequencing
ChOP-Seq	Chromatin Oligo affinity Rrecipitation Sequencing
circRNA	circular RNA
CLASH	Crosslinking, Ligation, and Sequencing of Hybrids
CLIP	Crosslinking Immunoprecipitation

CNCI	Coding-Non-Coding-Index
CNN	Convolutional Neural Network
DBD	DNA Binding Domains
DBS	DNA Binding Sites
DeepCPP	Deep neural network for RNA Coding Potential Prediction
DNN	Deep Neural Network
ENCODE	Encyclopedia of DNA Elements
FIS	Functional Information Score
FN	False Negative
FP	False Positive
FPKM	Fragment Per Kilobase per Million reads
FPR	False Positive Rate
GB	Gradient Boosting trees
GEO	Gene Expression Omnibus
GO	Gene Ontology
HM	Histone Modification
Hm	Harmonic mean
HTS	High Throughput Sequencing

lncIBTP	lncRNA Interaction Biomolecule Type Prediction
KNN	K-nearest Neighbor
LGBM	Light Gradient Boosting Machine
lincRNA	Long intergenic ncRNA
lncRNA	Long Non-coding RNA
LncRNA-ID	Long non-coding RNA IDentification
MCC	Matthew's Correlation Coefficient
mDS	minimum Distribution Similarity
miRNA	microRNA
MRE	MiRNA Responsive Elements
mRNA	Messenger RNA
NCBI	National Center for Biotechnology Information
ncRNA	non-coding RNA
NGS	Next-generation Sequencing
ORF	Open Reading Frame
PCC	Pearson Correlation Coefficients
piRNA	PIWI-interacting RNA
PRC	Precision Recall Curve

RAP-Seq	RNA Antisense Purification Sequencing
RPKM	Reads Per Kilobase of transcript, per Million mapped reads
siRNA	small interfering RNA
RefSeq	Reference Sequence
RF	Random Forest
RNA-Seq	RNA Sequencing
ROC	Receiver Operating Characteristic
rRNA	ribosomal RNA
Sn	Sensitivity
snRNA	small nuclear RNA
snoRNA	small nucleolar RNA
sORF	small Open Reading Frames
Sp	Specificity
SVM	Support Vector Machine
SWSS	Smith Waterman Subgraph Similarity
TDF	Triplex Domain Finder
TF	Transcription Factor
TFO	Triplex Forming Oligo

th	threshold
TN	True Negative
TP	True Positive
TPM	Transcripts Per Million
TPR	True Positive Rate
TriplexFPP	Triplex Forming Potential Prediction
tRNA	transfer RNA
UCSC	University of California Santa Cruz Genome Browser website
XCI	X-Chromosome Inactivation
XGB	XGBoost

List of Publications

Journal Publications

1. **Y. Zhang**, Y. Long, and C. K. Kwoh, “Class similarity network for coding and long non-coding RNA classification,” *BMC Bioinformatics*, vol. 22, no. 1, pp. 1–14, 2021.
2. F.Cao[†], **Y. Zhang**[†], Y. Cai[†], S. Animesh, Y. Zhang, S. Akincilar, *et al.* and M.J. Fullwood, ”Chromatin Interaction Neural Network (ChINN): a machine learning-based method for predicting chromatin interactions from DNA sequences,” *Genome Biology*, vol. 22, no. 226, 2021.
3. **Y. Zhang**[†], Y. Cai[†], X. Roca, M. J. Fullwood, and C. K. Kwoh, “Chromatin loop anchors predict transcript and exon usage,” *Briefings in Bioinformatics*, vol. 22, no. 6, bbab254, Nov 2021.
4. **Y. Zhang**, C. Jia, and C. K. Kwoh, “Predicting the interaction biomolecule types for lncRNA: an ensemble deep learning approach,” *Briefings in Bioinformatics*, vol. 22, no. 4, bbaa228, Jul 2021.
5. **Y. Zhang**, C. Jia, M. J. Fullwood, and C. K. Kwoh, “DeepCPP: a deep neural network based on nucleotide bias information and minimum distribution similarity feature selection for RNA coding potential prediction,” *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 2073–2084, 2021.
6. **Y. Zhang**, Y. Long, and C. K. Kwoh, “Deep learning based DNA: RNA triplex forming potential prediction,” *BMC Bioinformatics*, vol. 21, p. 522, 2020.

7. **Y. Zhang**, Y. Long, R. Yin, and C. K. Kwoh, "DL-CRISPR: a deep learning method for off-target activity prediction in CRISPR/Cas9 with data augmentation," *IEEE Access*, vol. 8, pp. 76610-76617, 2020.
8. Y. Long, **Y. Zhang**, M. Wu, S. Peng, C. K. Kwoh, J. Luo, and X. Li, "Heterogeneous graph attention networks for drug virus association prediction," *Methods*, 2021.
9. Y. Long, J. Luo, **Y. Zhang**, and Y. Xia, "Predicting human microbe–disease associations via graph attention networks with inductive matrix completion," *Briefings in Bioinformatics*, vol. 22, no. 3, bbaa146, May 2021.
10. R. Yin, E. Luusua, J. Dabrowski, **Y. Zhang**, and C. K. Kwoh, "Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks," *Bioinformatics*, vol. 36, no. 9, pp. 2697-2704, 2020.
11. R. Yin, **Y. Zhang**, X. Zhou, and C. K. Kwoh, "Time series computational prediction of vaccines for influenza a h3n2 with recurrent neural networks," *Journal of bioinformatics and computational biology*, vol. 18, no. 01, p. 2040002, 2020.
12. R. Yin, N. N. Thwin, P. Zhuang, **Y. Zhang**, Z. Lin, and C. K. Kwoh, "IAV-CNN: a 2D convolutional neural network model to predict antigenic variants of influenza A virus," *bioRxiv*, 2020.

Conference Proceedings

1. **Y. Zhang**, Z. Lin, and C. K. Kwoh, "Information theory-based feature selection: minimum distribution similarity with removed redundancy," In *International Conference on Computational Science*, pp. 3-17. Springer, Cham, 2020.
2. Y. Long, **Y. Zhang**, M. Wu, S. Peng, C. K. Kwoh, J. Luo, and X. Li, "Predicting drugs for COVID-19/SARS-CoV-2 via heterogeneous graph attention networks," In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 455-459. IEEE, 2020.

Chapter 1

Introduction

1.1 Background

1.1.1 Transcript, Exon, and Coding RNA

The transcription is the initial phase for gene expression. In this stage, the information encoded within the particular segment of DNA of one or more genes is transcribed into a strand of RNA (or pre-mRNA) [1]. The pre-mRNA can further produce multiple types of functional RNA molecules (or transcripts) like coding RNA and non-coding RNA (ncRNA) via different processing steps, where the coding RNA is also known as the messenger RNA (mRNA).

Exons refer to the segments of DNA that contain information coding for proteins or peptide sequences. In splicing, the introns (noncoding stretches) of the pre-mRNA are excluded and the exons are alternatively spliced to generate the mature mRNA molecules [2]. mRNAs act as mediators between DNA and protein.

1.1.2 Long Non-coding RNA (lncRNA)

Long Non-coding RNA (lncRNA) is an enormous class of RNA molecules whose transcript length is longer than two hundred nucleotides. Recently, the high-throughput sequencing

Fig. 1.1: The demonstration of the relationship among RNA transcript, exon, mRNA, and lncRNA.

(HTS) techniques have discovered thousands of lncRNAs. And the lncRNAs are found to make up a majority of the transcripts [3]. lncRNA variants are mainly generated by the excision of the first or last exon [4]. A brief overview of the relationship among RNA transcript, exon, mRNA, and lncRNA is illustrated in Fig.1.1.

According to the genome location to the nearest genes that encode to proteins, lncRNAs are broadly classified into five types, including sense lncRNA, antisense lncRNA, intronic lncRNA, long intergenic ncRNA (lincRNA), and bidirectional lncRNA. Sense and antisense lncRNAs span exons of genes that encode to proteins on the coding strands and the complementary strands separately. Intronic lncRNAs locate within the intronic regions of genes that encode to proteins. lincRNAs locate between two genes that encode to proteins. The transcriptions of bidirectional lncRNAs are opposite to the transcriptional initial locus of genes that encode to proteins [5, 6]. The representation of the genomic loci of different types of lncRNAs is shown in Fig.1.2.

1.1.3 The Role and Application of lncRNA

Previous studies show that lncRNAs play significant roles in diverse regulatory functions to affect gene expression in a variety of physiopathological and biological contexts. Along with other biomolecules like DNA, RNA, and proteins, lncRNAs take part in the localization to target loci, the coordination of regulatory proteins, modeling the three-dimensional nuclear organization, *etc* [7–9], to carry out their wide range of functions such as modifying epigenetic, altering the splicing, stability, and translation of the gene,

Fig. 1.2: The subtypes of lncRNAs grouped by the genome location to the nearest protein-coding genes. Sense lncRNAs and antisense lncRNAs cross exons of genes that can encode to proteins on the coding strands and complementary strands separately, intronic lncRNAs locate within the intronic regions of genes that can encode to proteins, long intergenic ncRNAs (lincRNAs) locate between 2 genes that encode to proteins, and bidirectional lncRNAs transcribed opposite to the transcriptional initial locus of protein-coding genes.

modulating the structure, state, and function of chromatin, and so on [10–14]. lncRNAs also are found to regulate the formation of organelles and nuclear condensates [15].

Besides, accumulating pieces of evidence suggest that lncRNA are associated with all kinds of diseases like diabetes, cancer, neurodevelopment and psychiatric disease, and so on [16–18]. The expression patterns of lncRNA, *i.e.* condition-specific and tissue-specific, indicate that they could provide potential therapeutic applications as therapeutic targets and diagnostic markers [19]. For example, lncRNA *MEG3* was found to be related to the tumor size and lymph node metastasis and its expression was shown to be associated to the cervical cancer [20], lncRNA *GACAT3* was observed to be upregulated in lung cancer tissues and might be used as a molecular marker for lung cancer [21], lncRNA *BANCR* could serve as a biomarker and therapeutic target of breast cancer by inhibiting the related process and expression to inhibit the invasion, proliferation, and metastasis of breast cancer cells [22], and lncRNA *H19* highly expressed in atherosclerosis which makes it a potential therapeutic target for atherosclerosis [23].

1.2 Motivation and Objective

Although functional studies suggest crucial roles for lncRNAs, only a few of lncRNAs are well studied, most of the lncRNAs await further verification [24].

Accordingly, lncRNA functional annotation mainly includes two parts: 1) lncRNA annotation, that is distinguishing a lncRNA transcript from other types of transcripts, and 2) lncRNA function exploration, which is identifying the functional structure of a lncRNA along with its interacting biomolecules and the corresponding binding sites [25].

The data source to identify transcript comes from HTS experiments [26]. Several steps are included for lncRNA functional annotation with the experimental approaches. The RNA sequences are assembled according to the sequencing reads from HTS experiments. The RNA sequences undergo further analysis to be annotated as different groups of RNA transcripts. And *in vivo* or *in vitro* experiments are executed to identify the lncRNA functions.

However, experimental approaches for annotating transcripts and identifying their functions are expensive and time-consuming. Moreover, lncRNAs have low conservation levels and low expression levels, as well as high biogenesis diversity compared to mRNAs [3], which pose challenges to their function exploration. Considering these factors, the *in silico* methods would be favorable choices for lncRNA functional annotation.

In this thesis, I seek to present machine learning approaches for lncRNA function annotation. I start by predicting the transcripts using data obtained from HTS experiments, then I develop an *in silico* method to distinguish lncRNA from mRNA. After that, I present a computational approach to predict the types of biomolecules that a lncRNA potentially interacts with, and then focus on one specific type of lncRNA-DNA interaction, *i.e.* DNA:lncRNA triplexes to help the comprehensions of lncRNA functions. The questions in lncRNA functional annotation and the objectives of this thesis are briefly shown in Fig.1.3.

1.3 Major Contributions

This thesis mainly focuses on the functional annotation of lncRNAs using machine learning models. To serve this purpose, I start by predicting the RNA transcripts from DNA, followed by distinguishing the lncRNA transcript from coding transcripts, then I predict the possible types of interacting biomolecules for lncRNAs, finally, I focus on one specific interaction mechanism to predict the potential lncRNAs that can form triplex with DNA. The major contributions of this thesis are listed below.

Fig. 1.3: The problem formulation in lncRNA function annotation and the objectives in this thesis.

- Predicting the transcript and exon usage from transcription factors and ChIA-PET (Chromatin Interaction Analysis with Paired-end Tags) data which indicates locations of chromatin loops in the genome. I do not just confine ourselves to splice sites, in contrast to previous splicing event prediction methods which use the level of splicing calculated from the exonic expression level involved in that splicing event. I predict the exon usage independently and quantified the transcript directly. Our model performs well on independent test datasets, which indicates its robustness. Examining the predictive features, I find that RNA Polymerase II ChIA-PET data is one of the most important features in both transcript and exon usage prediction, suggesting that chromatin loop anchors are predictive of both transcript and exon usage.
- Proposing a novel feature representation approach to extract information around the start codon for the most possible coding region and developing a deep framework named DeepCPP (Deep neural network for RNA Coding Potential Prediction) to distinguish lncRNA and mRNA in multi-species. I address the issue of distinguishing lncRNA and mRNA with small Opening Reading Frames (sORF), which used to be a bottleneck. Our newly proposed feature representation method works effectively on sORF type data. And the DeepCPP is proved to perform better than other predictors developed in recent years on *human*, vertebrate, and insect species, especially on transcript with sORF.

- Developing lncIBTP (lncRNA Interaction Biomolecule Type Prediction) to identify the types of biomolecules that a lncRNA potentially associated with. This should be the first attempt to predict the interaction biomolecule types for a lncRNA. I illustrate that the types of biomolecules interacting with lncRNAs could be successfully predicted.
- Introducing TriplexFPP (Triplex Forming Potential Prediction) to predict the lncRNAs and potentials of DNA sites to form DNA:lncRNA triplexes in practice using data from *in vivo* or *in vitro* assays. No machine learning programs have been proposed in predicting DNA:lncRNA triplex-forming potential according to our best knowledge. I show that TriplexFPP works effectively in DNA:lncRNA triplex-forming potential prediction.

1.4 Thesis Organization

The remaining parts of this thesis are arranged as follows. Chapter 2 reviews the experimental methods, databases, and computational methods for lncRNA functional annotation. Chapter 3 discusses the prediction of transcription and exon usage from transcription factors and ChIA-PET data using a machine learning method. Chapter 4 introduces DeepCPP to distinguish lncRNA from mRNA on diverse species with a novel feature representation method. Chapter 5 and Chapter 6 focus on the interactions of lncRNAs to decipher the lncRNA functions by presenting lncIBTP which predicting the types of biomolecules that a lncRNA would interact with and TriplexFPP which predicting the DNA:lncRNA triplex-forming potential prediction. Chapter 7 concludes the thesis and points out the potential future directions.

Chapter 2

Literature Review

This chapter delivers a literature review about lncRNA functional annotation. Firstly, the experimental methods are reviewed. Secondly, the data type and databases are briefly described. Thirdly, the computational methods are reviewed.

2.1 Experimental Methods

Next-generation sequencing (NGS), also known as high-throughput sequencing (HTS), enables fast and cheap sequencing of DNA and RNA. NGS plays a dominant role in the area of genomics and molecular biology. This section briefly introduces the techniques related to the RNA transcript discovery and lncRNA function determination that will be used as the data source in later chapters.

RNA sequencing (RNA-Seq) is the sequencing approach that utilizes NGS technology to map, quantify and analyze the transcriptome. It also delivers the information on the expression levels of the transcripts and the alternate splice variants [27]. RNA-Seq is usually used to discover novel transcripts.

Chromatin immunoprecipitation sequencing (ChIP-Seq) is a popular technique for analyzing epigenetic histone modifications and binding sites for proteins. It is used to identify local posttranslational modifications of the histone tails and to monitor changes in the modifications in response to different stimuli [28]. ChIP-Seq can be used together with RNA-Seq to unravel the transcriptional regulatory network [29].

Chromatin interaction analysis with paired-end tags (ChIA-PET) provides the information of chromatin interactions that are relevant to interested proteins in the genome. It reveals genome-wide chromatin interactions by incorporating chromatin proximity ligation, chromatin ChIP-based enrichment, sonication-based chromatin fragmentation, and Paired-End Tag sequencing [30].

The lncRNA function study mainly depends on the identification of interacting biomolecules. Such interactions including lncRNA with RNA, DNA, and protein/chromatin, which can be detected by NGS techniques like Chromatin Isolation by RNA Purification Sequencing (ChIRP-Seq) [31], Capture Hybridization Analysis of RNA Targets Sequencing (CHART-Seq) [32], RNA Antisense Purification Sequencing (RAP-Seq) [33], and Chromatin Oligo affinity Precipitation Sequencing (ChOP-Seq) [34], Crosslinking, Ligation, and Sequencing of Hybrids (CLASH), Crosslinking Immunoprecipitation (CLIP) [35,36], *etc.*

2.2 Databases

This section briefly introduces three main genome browsers and the corresponding databases that will be used in this paper.

2.2.1 National Center for Biotechnology Information (NCBI)

National Center for Biotechnology Information (NCBI, <https://www.ncbi.nlm.nih.gov/>) aims to facilitate the understanding of basic genetic and molecular processes that influence disease and health. It provides a series of databases relevant to biotechnology and biomedicine.

The Gene Expression Omnibus (GEO, <http://www.ncbi.nlm.nih.gov/geo/>) is a database built and maintained at NCBI. It is an international public repository that aims to provide the next-generation sequence functional genomic datasets and high-throughput microarray datasets. It contains next-generation sequence data that examine quantitative gene expression and regulation, epigenomics, and other functional genomics-related data using methods like RNA-Seq, ChIP-Seq, and so on [37].

The Reference Sequence (RefSeq) database is another database developed by NCBI to serve purposes like genome annotation and sequence variation location reporting in medical records [38]. It provides an integrated, comprehensive, well-annotated, and non-redundant set of sequences for DNA, RNA, and protein, along with the separate and linked records arising from them [38, 39].

2.2.2 University of California Santa Cruz Genome Browser website (UCSC)

The University of California Santa Cruz Genome Browser website (UCSC, <https://genome.ucsc.edu>) is a viewer for genome annotations. It provides a variety of annotation datasets, including gene-expression data, gene predictions, mRNA alignments, disease-association data, mappings of DNA repeat elements, and mappings of commercially available gene chips [40].

The Encyclopedia of DNA Elements (ENCODE, <https://www.encodeproject.org/>) is a public research project focused on the human genome. It aims to identify those functional components. For example, the elements that regulate cells and circumstances when a gene is active, and the elements that execute functions at the level of RNA and protein [41]. Its phase I analyzes a defined portion of the human genome sequence, phase II analyzes the entire genome, and phase III broadens the investigations, such as the repertoires of DNA methylation in cell and tissue, structure and modification of chromatin, chromatin looping, RNA transcription, and occupancy by RNA-binding proteins and transcription factors [42].

GENCODE (<https://www.gencodegenes.org/>) is a part of the ENCODE project. Its goal is to identify all gene features on the basis of evidence in the human genome with high accuracy via the process of manual curation, targeted experimental approaches, and different computational analysis [43].

2.2.3 Ensembl

Ensembl (<https://www.ensembl.org>) is a genome browser for vertebrate genomes. It supports comparative genomics, variation, and regulatory data. Ensembl feeds sequence

Table 2.1: The three genome browsers that will be used in this thesis.

Browser	url	Database	Description
NCBI	https://www.ncbi.nlm.nih.gov	GEO	Gene expression profiling and RNA methylation database
		RefSeq	Curated non-redundant sequence database of genomes
UCSC	https://genome.ucsc.edu	ENCODE	Human whole-genome database
		GENCODE	Encyclopedia of genes and gene variants
Ensembl	www.ensembl.org		Genome browser for vertebrate genomes

data into the gene annotation system and thus creating a set of predicted gene locations for subsequent analysis and display [44].

A summary of the above three genome browsers is listed in Table 2.1.

2.3 Computation Methods

2.3.1 RNA Transcript and Exon Usage Prediction

Epigenomics and transcriptomics data from HTS techniques such as ChIP-Seq and RNA-Seq are popular in predicting gene transcript expression due to their dominant roles in revealing the expression levels of genetic features and protein-DNA binding. Recently, machine learning has gained widespread attention and has been successfully applied to predict splicing using data from RNA-Seq and/or ChIP-Seq experiments, or DNA sequences. Leung *et al.* present a model trained with RNA-Seq data to predict splicing events in a tissue level in *mouse*, both in individual and across the tissues [45], Jha *et al.* propose a transfer learning-based framework by incorporating CLIP-Seq, knock-down, and overexpression experiments in *mouse* tissues [46], a computational framework named DARTS has been developed to reveal differential splicing events across biological samples using a deep learning model combined with evidence from empirical RNA-Seq experiments [47], and studies such as Epigenome-based Splicing Prediction also consider

other types of data like DNase-Seq, eCLIP, methylation, MNase-Seq and build a Recurrent Neural Network [48], which use adjacent epigenetic signals around the splice sites. A summary of these prediction methods is listed in Table 2.2.

2.3.2 lncRNA and mRNA Classification

The HTS techniques enable numerous transcripts to be discovered. However, distinguishing lncRNAs from mRNAs still is a challenging task due to the similar features shared between them, for example, splicing, poly(A) tails, and comparable sequence length [49]. To solve this issue, some computation methods are developed.

In the early stage, alignment-based methods like CPC [50] and PhyloCSF [51] have been proposed based on existing databases. But these tools are time-intensive and limited by previous knowledge of the data. Later on, alignment-free methods have been developed which only use the intrinsic sequence features. For example, CPAT is constructed with a logistic regression model using four types of sequence features [52], CNCI is reported to effectively classify sense-antisense and incomplete transcripts pairs by profiling adjoining nucleotide triplets [53], PLEK is proposed using an improved k-mer method as well as the sliding window strategy to examine the transcripts [54], LncRScan-SVM is developed based on analysing the qualities of the nucleotide and codon sequences, and also the structures of transcripts [55], LncRNA-ID is proposed by taking advantage of features of the alignments dependent on the profile hidden Markov model [56], CPC2 is updated over CPC which are faster and more accurate [57], COME is presented as a robust tool for the prediction and characterization of novel lncRNAs supported by multiple-feature [58], FEELnc provides a standardized and complete solution to identify lncRNAs from transcripts obtained from RNA-Seq experiments [59], an SVM method is presented using the features of the nucleotide frequencies selected by PCA and the length corresponding to the first ORF [60], LncFinder is built on SVM model which uncovers the features of lncRNA and mRNA from diverse aspects [61], and the CPPred is reported to be able to distinguish between small ncRNAs and the small mRNAs [62]. Recently, deep learning models have also been adopted in distinguishing mRNA and lncRNA, which show varied efficiencies for different datasets. For example, DeepLNC [63] and RNAsamba [64]

Table 2.2: The approaches in alternative splicing prediction discussed in this section.

Method	Cell line or tissue	Data type	Model	Target	Reference
Leung <i>et al.</i>	Mouse tissue: Heart, kidney, brain, liver, testis	RNA-Seq	DNN	Splicing event: low, medium and high percent-spliced-in (PSI) value for a particular tissue type (bootstrap)	[45]
Jha <i>et al.</i>	Mouse tissue: Heart, kidney, brain, liver, testis	RNA-Seq, CLIP-Seq	BNN, DNN	Splicing event: low, medium and high (PSI) value for a particular tissue type (AVISPA)	[46]
DARTS	HepG2 and K562	RNA-Seq, DNA sequences,	DNN	Splicing event: unchanged or differential splicing events with high- confidence between conditions (Bayesian hypothesis testing)	[47]
ESPRNN	Breast Myoepithelial Cells, Breast vHMEC, Adult Liver, Lung, A549, GM12878, HEPG2, HMEC, K562	RNA-Seq, ChIP-Seq, eCLIP, DNase-Seq, MNase-Seq, methylations (WGBS and RRBS), DNA sequences	RNN	Splicing event: level of splicing (inclusion and skipping for exon are determined with PSI cutoffs (20% and 80%) the exon inclusion is determined by binarizing the exonic expression level with a threshold of 1)	[48]

are built on the top of deep neural network (DNN), mRNN is constructed based on the recurrent neural network (RNN) [65], LncADeep is trained using the deep belief network [66], LncRNAet is developed by integrating the RNN and convolutional neural network (CNN) [67], *etc.* The computational methods in distinguishing mRNA and lncRNA are briefly summarized in Table 2.3.

2.3.3 lncRNA Function Prediction

Although a huge amount of lncRNAs have been annotated, few of them are well characterized regarding the functions. Experimental approaches to reveal lncRNA function are expensive and time-consuming. Specifically, the relatively low expression level of lncRNAs poses further challenges for experimental approaches. Therefore, computational methods are favored ways to predict the lncRNA functions.

Some lncRNA function prediction approaches rely on the calculation of the similarities between lncRNA and the known RNA or protein. These methods assume that co-expressed genes with similar expression patterns may share similar functions in related processes [68]. Such kind methods usually use experimental expression data like microarray and HTS data to infer the functions of lncRNAs according to the neighboring genes indicated by the Gene Ontology (GO) terms. For example, lncRNA2Function maps lncRNA genes to biological pathways and GO terms [69], KATZLGO annotates lncRNAs according to neighboring protein-coding genes by calculating their similarities with proteins [70], and lncFunTK annotates the functions of lncRNAs through the regulatory network constructed from the integrated high throughput sequencing data [71].

Other lncRNA function prediction approaches focus on finding the possible interactions between lncRNA and certain biomolecules. Such interactions mainly include lncRNA-RNA, lncRNA-Protein, and lncRNA-DNA. For example, PmliPred and CIRNN are developed to predict the interaction in lncRNA-miRNA using deep learning models [72, 73]; HLPI-Ensemble, SFPEL-LPI, and PLRPIM are presented to predict the interaction between lncRNA and protein using sequence-based features [74–76]; DeepLGP is proposed to prioritize lncRNA target genes [77]; Wang *et al.* construct deep learning frameworks to infer the binding locus for 12 lncRNAs throughout the genome according to the ChIRP-Seq [78]; tools like Triplexator [79], LongTarget [80], Triplex Domain

Table 2.3: The computational methods in lncRNA and mRNA classification discussed in this section.

Program	Method	Year	Reference
Coding Potential Calculator (CPC)	SVM	2007	[50]
PhyloCSF	Phylogenetic Model	2011	[51]
Coding Potential Assessment Tool (CPAT)	Logistic Regression Model	2013	[52]
Coding-Non-Coding-Index (CNCI)	SVM	2013	[53]
Predictor of Long non-coding RNAs and mEssenger RNAs based on an improved K-mer scheme (PLEK)	SVM	2014	[54]
LncRScan-SVM	SVM	2015	[55]
Long non-coding RNA IDentification (LncRNA-ID)	Random Forest	2015	[56]
DeepLNC	DNN	2016	[63]
CPC2	SVM	2017	[57]
COding potential calculator based on Multiple Evidences (COME)	Random Forest	2017	[58]
FlExible Extraction of LncRNAs (FEELnc)	Random Forest	2017	[59]
Hugo's SVM	SVM	2017	[60]
mRNA RNN (mRNN)	RNN	2018	[65]
LncRNAnet	RNN & CNN	2018	[67]
LncADeep	Deep Belief Network	2018	[66]
LncFinder	SVM	2019	[61]
Coding Potential Prediction tool (CPPred)	SVM	2019	[62]
RNASamba	DNN	2020	[64]

Finder (TDF) [81], and Triplex-Inspector [82] statistically compute the RNA-DNA triple helix formation based on the Hoogsteen and reverse-Hoogsteen base-pairing.

Table 2.4 briefly summarized the above mentioned lncRNA function prediction approaches.

2.4 Evaluation Matrix

This section introduces diverse evaluation criteria which will be used in the following chapters. For two-class classification problems, we use the criteria of accuracy (Acc), sensitivity (Sn) (or Recall), specificity (Sp), F-score, harmonic mean (Hm) of Sn and Sp, Matthew's correlation coefficient (Mcc), Area Under Receiver Operating Characteristic curve (AUROC), and Area Under Precision-Recall Curves (AUPRC) to evaluate the model performances. The equation to calculate Acc, Sn, Sp, F-score, Hm of Sn and Sp, and Mcc are listed below.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

$$Sn/recall = \frac{TP}{TP + FN} \quad (2.2)$$

$$Sp = \frac{TN}{TN + FP} \quad (2.3)$$

$$precision = \frac{TP}{TP + FP} \quad (2.4)$$

Table 2.4: The lncRNA function prediction approaches discussed in this section.

Program	Interaction molecule	Data Type	Core algorithm	Year	Reference
LncRNA2Function	mRNA	RNA-Seq	Pearson Correlation Coefficients (PCC)	2015	[69]
lncFunTK	miRNA	RNA-Seq, CLIP-Seq, ChIP-Seq	Functional Information Score (FIS)	2018	[71]
CIRNN	miRNA	Sequence	CNN and independently recurrent neural network (IndRNN)	2019	[73]
PmliPred	miRNA	Sequence	CNN-BiGRU	2020	[72]
HLPI-Ensemble	Protein	Sequence	Ensemble of SVM, RF, and XGB, respectively	2018	[74]
SFPEL-LPI	Protein	Sequence	Smith Waterman subgraph similarity (SWSS)	2018	[75]
KATZLGO	Protein	expression profiles	KATZ measure	2019	[70]
PLRPIM	Protein	Sequence	RF and light gradient boosting machine (LGBM)	2019	[76]
Triplexator	DNA	Sequence	Automaton-based pre-filter and exhaustive search	2012	[79]
Triplex-Inspector	DNA	Sequence	Exhaustive search and examine uniqueness	2013	[82]
LongTarget	DNA	Sequence	24 rule-sets use overlapping	2015	[80]
TDF	DNA	Sequence	triplex-forming oligo (TFOs) to find contiguous regions within the RNA	2019	[81]
DeepLGP	DNA	Gene expression and location in the genome	Graph convolutional network	2020	[77]

$$F - score = \frac{2 \times precision \times Sn}{precision + Sn} \quad (2.5)$$

$$Hm = \frac{2 \times Sn \times Sp}{Sn + Sp} \quad (2.6)$$

$$Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}} \quad (2.7)$$

where TP, FN, TN, and FP refer to the true positive, false negative, true negative, and false positive, respectively.

For four-class classification problems, we use the criteria of overall Acc, the macro-average F-score, the macro-average precision, and the macro-average recall. The calculations of Acc, precision, and recall are the same as that listed in the two-class classification problem above, but the TP and FP refer to the amount of correctly and wrongly predicted samples in the interested class, and TN and FN represent the amount of correctly and wrongly predicted samples in the other 3 classes. The macro-average precision, recall, and F-score are calculated as:

$$Macro - precision = \frac{\sum_i precision}{4}, i = 1, 2, 3, 4 \quad (2.8)$$

$$Macro - recall = \frac{\sum_i recall}{4}, i = 1, 2, 3, 4 \quad (2.9)$$

$$Macro - F - score = \frac{2 \times (Macro - precision) \times (Macro - recall)}{(Macro - precision) \times (Macro - recall)}, i = 1, 2, 3, 4 \quad (2.10)$$

2.5 Chapter Summary

This chapter presents a literature review of the related methods in lncRNA functional annotation. We first introduce the experimental methods which provide the raw data for the RNA transcript discovery and lncRNA function determination. The experimental methods are mainly NGS techniques, including RNA-Seq, ChIP-Seq, ChIA-PET, *etc.* Next, three main genome browsers and the corresponding databases are introduced. These databases integrate data obtained from a variety of HTS experiments as well as the sequences of DNA, RNA, and protein, which are the main data sources for the studies in later chapters. Although the transcript annotation and lncRNA function determination can be obtained from the *in vivo* or *in vitro* assays, these experimental methods usually are costly and time-intensive. Therefore, fast and accurate *in silico* approaches are needed. We also review the available computational tools for transcription prediction, lncRNA and mRNA classification, and lncRNA function prediction. In the following chapters, we will discuss the limitations of the existing *in silico* approaches and propose new methods to solve the current challenges.

Chapter 3

Chromatin Loop Anchors Predict Transcript and Exon Usage¹

3.1 Introduction

Transcription is the synthesis of RNA from DNA. An exon is defined as a gene region that will be coded to proteins or peptide sequences. Differential exon usage via alternative splicing can lead to a variety of gene products from the genome [84]. More specifically, the usage of exons has the potential to alter the stability, localization, or translation of RNAs, and the localization, efficiency, specificity, or life cycle of proteins [85]. Dysregulation in gene transcription and exon usage is related to disease [86–89]. For example, certain isoforms of PKM2 and MAP2 gene have higher expression in the later stage of neuroblastoma compared with other isoforms [88]. This preference of certain isoforms may be a result of tumorigenesis and therefore exon usage in cancer can reveal more information about cancer pathways.

RNA sequencing (RNA-Seq) plays a dominant part in revealing the expression levels of genes, transcripts, or exons between samples [90]. RNA-Seq can be adopted to quantify exon usage by meaning the number of reads that fall within a particular exon. Additionally, RNA-Seq data can be employed to quantify alternative splicing events by

¹The work in this chapter has been published in [83].

using the junction reads of RNA-Seq data. For example, Kakaradov *et al.* estimate the level of alternative splicing by developing probabilistic models based on position-specific read counts from RNA-Seq data [91].

Exon usage or the percent inclusion of alternatively spliced junctions obtained from RNA-Seq can be used to interrogate the question of alternative splicing depending on the nature of the question studied. Lee *et al.* used exon usage. Specifically, Lee *et al.* measured the splicing events according to exonic expression level, which is measured by fragment per kilobase per million reads (FPKM) obtained from RNA-Seq [48]. By contrast, some methods used percent inclusion. For example, Goldstein *et al.* assembled the splice junctions and exons in terms of mapped reads into a genome-wide splice graph and identified the splice events from the graph [92]. Zhang *et al.* revealed the differential splicing events using a Bayesian statistical model by considering the evidence from specific RNA-Seq experiments and the prior probability of splice events [47].

Chromatin Immunoprecipitation Sequencing (ChIP-Seq) facilitates the identification of localization of protein–DNA binding sites throughout the genome [93]. ChIP-Seq produces transcription factors (TFs) data on gene expression. Histone modification (HM) regulates gene expression via altering recruiting histone modifiers which is a type of important chemical reaction at the chromatin or altering chromatin structure [94]. ChIP-Seq peaks for TFs and transcriptome data are used individually or in a combinatorial manner for gene regulation and expression analysis or prediction. For example, BETA combined ChIP-Seq and transcriptome data to unravel the regulation of gene expression [95], and DeepDiff interpreted how the differential patterns of gene regulation are affected by the dependencies among HMs [94]. Different kinds of transcription factor synthetic indexes extracted from ChIP-Seq data are adopted to identify the gene expression levels [96] and predict TF bindings [97]. Moreover, some studies integrate the data from both RNA-Seq and ChIP-Seq assays to reveal the relationships between specific HMs and various aspects of differential splicing. For example, Hu *et al.* developed computational methods to reveal how histone posttranslational modifications may affect the alternative splicing in mammalian brain [98].

Recently, machine learning has gained widespread attention and has been successfully applied to predict splicing using data from RNA-Seq and/or ChIP-Seq experiments,

or DNA sequences. Leung *et al.* presented a model trained with RNA-Seq data to predict splicing events at a tissue level in *mouse*, both in individual and across the tissues [45], Jha *et al.* proposed a transfer learning-based framework by incorporating CLIP-Seq, knockdown, and over-expression experiments in *mouse* tissues [46], a computational framework named DARTS was developed to reveal differential splicing events across biological samples using a deep learning model combined with evidence from empirical RNA-Seq experiments [47].

Studies such as Epigenome-based Splicing Prediction using a Recurrent Neural Network (ESPRNN) also considered other types of data like DNase-Seq, eCLIP, methylation, MNase-Seq [48]. However, ESPRNN only used adjacent epigenetic signals around the splice sites, but not across the entire gene. Besides, none of these methods considered any chromatin interaction-related data, *i.e.* Chromatin Interaction Analysis by Paired-End Tag Sequencing (ChIA-PET) data.

ChIA-PET is a technique to determine *de novo* long-range chromatin interactions throughout the genome, it incorporates chromatin proximity ligation, chromatin ChIP-based enrichment, sonication-based chromatin fragmentation, and Paired-End Tag sequencing [99]. Chromatin interactions can regulate gene transcription [100]. Oncogenes and remote regulatory elements can be brought into close spatial proximity through chromatin interactions [101], which may serve as a structural basis for protein binding and thus leading to transcription regulation. Chromatin organization is correlated with alternative splicing [102]. A recent study demonstrated that the degradation of RNA polymerases can affect local chromatin architectures and these local regions include high RNA polymerase binding sites and active promoters [103]. However, no other studies report on whether chromatin interactions have any ability to predict exon usage.

On the basis of the above-mentioned evidence, we hypothesized that apart from the RNA-Seq and ChIP-Seq data, ChIA-PET data indicating chromatin loop anchors might also be of predictive value, but it has not been tested in transcript and exon usage prediction. Hence, here we asked the question: can we apply ChIA-PET data to transcript and exon usage prediction, and will such data contribute to the prediction? What are the chromatin factors that can predict transcript and exon usage, if any? Therefore, in this chapter, we expected to use a machine learning approach, *i.e.* gradient boosting

trees (GB), to predict the transcript and exon usage using the ChIA-PET data as well as transcription factor data.

As opposed to previous splicing event prediction methods which use the level of splicing calculated from the exonic expression level involved in that splicing event, we predicted exon usage independently and quantified transcripts directly. Previous prediction methods mainly detect the most common type of alternative splicing event by predicting the alternate usages (inclusion or exclusion of exons) of the cassette exons. But we should note that there are other events (for example, stand-alone expression of a single exon) that will affect exon usage. Our exon usage measurements include cassette exons, 5' and 3' terminal exons, mutually exclusive exons, and even single-exon (intronless) transcripts. As far as we can tell, machine learning models have not been used to indicate the usage level for single exon transcripts.

Another difference between previous methods and ours is that previous methods such as ESPRNN use epigenomic signals adjacent to splice sites and DNA sequences, whereas we use epigenomic signals present throughout the gene, not just at splice sites. We collected and integrated the RNA-Seq, ChIP-Seq, ChIA-PET, and DNase-Seq data from different sources for GM12878, HeLaS3, and K562 cell lines. Then we developed GB models to see if we can apply these data successfully to different prediction tasks, including verified transcript prediction, transcription abundance prediction, and exon usage prediction. The workflow to conduct this study is illustrated in Fig.3.1.

We evaluated our models via 10-fold cross-validation, as well as the chromosome-split and cross-cell validations which could break the dependencies between training and test data. Results show that our model realizes accuracy larger than 0.7707 (average), 0.7755, and 0.7512 for verified transcripts prediction and 0.7700 (average), 0.7459, and 0.7463 for exon usage prediction, respectively, regarding the above three validation methods in different cell lines.

The validation results here illustrate the model's effectiveness and robustness, as well as the plausibility to predict the transcript and exon usage. Interestingly, we noticed that the ChIA-PET and chromatin-related transcription factor binding motif-related features could play more significant roles than some of the most frequently used ChIP-Seq features in both the transcription and exon usage prediction. Our results suggested that there may be a connection between chromatin factors and exon usage.

Fig. 3.1: The workflow to conduct the study.

3.2 Materials and Methods

3.2.1 Data Collection and Definition

We collected data from three cell lines including K562, GM12878, and HeLaS3 which contain all desired data from different experiments. A brief explanation of the three cell lines can be found in Table 3.1, where the three cell lines are from different tissues and have different karyotypes. We obtained all the publicly available datasets of three cell lines from the ENCODE consortium [41] which have both high-quality HM ChIP-Seq as well as high-quality ChIA-PET information. GENCODE hg19 GTF file was used as the reference annotation for transcripts and exons. The scrambled GTF was generated by randomly shuffling the coordinates and accession ID in the reference GTF file. This scrambled GTF was used to generate a negative dataset for transcript and exon matrices.

After that, we generated the feature matrices of transcript and exon for both positive and negative datasets.

3.2.1.1 Transcript Prediction Feature Matrices

- Transcript abundance. Nucleus longPolyA RNA-Seq data of K562, GM12878, and HeLaS3 cells were downloaded from GEO (GSM765387) and quantified using kallisto (0.45.1). We built two kallisto indexes using positive and negative GTF files. This resulted in a correctly quantified transcript abundance and a scrambled negative transcript abundance. We classified the transcript abundance into 4 categories according to their quartiles: not_transcribed, lowly_transcribed, transcribed, and highly_transcribed according to the TPM (Transcripts Per Million). TPM is a normalization method for RNA-Seq, it is calculated following the steps of i) dividing the read counts by the length of each gene in kilobases, ii) counting the values of all the reads per kilobase (RPK) in a sample and dividing them by 1,000,000, and iii) dividing the RPK values by the “per million” scaling factor. In each cell line, we sorted the unique TPM values from smallest to largest, then we found the 1st quartile (Q1), the 2nd quartile (Q2) (median), and the 3rd quartile (Q3) correspondingly. We defined the transcription with a TPM value range from smallest

Table 3.1: A summary of the cell lines used in this work.

Cell	Tissue	Karyotype	Description
GM12878	blood	normal	B-lymphocyte
HeLaS3	cervix	cancer	cervical carcinoma
K562	blood	cancer	leukemia

to Q1 as not_transcribed, Q1 to Q2 as lowly_transcribed, Q2 to Q3 as transcribed, and Q3 to largest as highly_transcribed. The value of Q1, Q2, and Q3 for three cell lines were recorded in Table 3.2.

- Transcription factor and HM binding. ChIP-Seq peaks of different TFs and HMs were downloaded from UCSC and overlapped with the transcript in the matrix. ChIP-Seq peaks are the genomic regions with significantly more enrichment for proteins or HMs compared with genomic background, they are normally several hundred to several kb long.
- DNase binding sites. DNase binding sites were downloaded from UCSC and overlapped with the transcript in the matrix.
- Chromatin interaction abundance. RNA Polymerase II (Pol2A) (GM12878 and K562) or CTCF (GM12878 and HeLaS3) ChIA-PET data was downloaded from ENCODE. The features are counted by overlapping with transcripts in the matrix. The number of ChIA-PET interactions that have either end overlapping with the transcript was used as the feature of chromatin interaction abundance.

We performed two prediction tasks using the data collected in this part. The first one is to predict verified transcripts as a two classes classification problem, and the other one is to predict the transcript abundance as a four classes classification problem. The corresponding sizes of positive and negative datasets in each class for three cell lines were recorded in Table 3.3.

Table 3.2: TPM value to separate transcribed level. Q1, Q2, and Q3 represent the first, the second (median), and the third quartile in sorted TPM values, respectively.

	smallest	Q1	Q2	Q3	largest
GM12878	0	0.0917	0.4550	2.0810	27598
HeLaS3	0	0.1060	0.5650	2.6560	27598
K562	0	0.0908	0.4910	2.2760	27598

3.2.1.2 Exon Usage Prediction Feature Matrices

Similarly, exon feature matrices were prepared following the process of transcript feature matrices.

- Exon usage coefficient. We first used STAR (2.7.3a) to map the RNA-Seq reads using the true hg19 fasta and GTF file. The alignment outcomes of the RNA-Seq data were further prepared and counted for the number of reads mapped to different exons using DEXseq helping scripts. Then, DEXseq (1.28.0) was used to inter coefficient of the exon usage in K562, GM12878, and HeLaS3 cells, where the relative usage of an exon is obtained as dividing the number of transcripts that contain this exon by the number of all transcripts from the same gene.
- Transcription factor and HM binding. ChIP-Seq peaks of different TFs and HMs were downloaded from UCSC and overlapped with the transcript in the matrix.
- DNase binding sites. DNase binding sites were downloaded from UCSC and overlapped with the transcript in the matrix.
- Chromatin interaction abundance. Pol2A (GM12878 and K562) or CTCF (GM12878 and HeLaS3) ChIA-PET data was downloaded from ENCODE. The features are counted by overlapping with transcripts in the matrix. The number of ChIA-PET interactions that have either end overlapping with the transcript was used as the feature of chromatin interaction abundance.

The corresponding data amounts in each class for three cell lines in predicting the exon usage were recorded in Table 3.4 and the features used in each cell line for both transcription and exon usage prediction were listed in Table 3.5.

Table 3.3: The data amount in transcription prediction.

		GM12878	HeLaS3	K562
Verified transcript prediction	Postive	126707	121848	122585
	Negative	126707	121848	122585
Transcription abundance prediction	Not_transcribed	224130	269975	244787
	Lowly_transcribed	58594	47116	53442
	Transcribed	117172	94255	106857
	Highly_transcribed	58620	47170	53430

Table 3.4: The data amount in exon usage prediction.

	Total	Positive (non-nan)	Negative (nan)
GM1287	746421	191334	555087
HeLaS3	746421	178054	568367
K562	746421	163581	582840

Table 3.5: The list of features in each cell line.

GM12878	HeLaS3	562
ChIAPET_Pol2A		ChIAPET_Pol2A
ChIAPET_CTCF	ChIAPET_CTCF	
Cebp	Cebp	Cebp
Cfos	Cfos	Cfos
Chd2	Chd2	Chd2
Cmyc	Cmyc	Cmyc
Corest	Corest	Corest
Ctcf	Ctcf	Ctcf
Dnase	Dnase	Dnase
E2f4	E2f4	E2f4
Elk	Elk	Elk
Ezh2	Ezh2	Ezh2
Gabp	Gabp	Gabp
H3K27ac	H3K27ac	H3K27ac
H3K27me3	H3K27me3	H3K27me3
H3K36me3	H3K36me3	H3K36me3
H3K4me2	H3K4me2	H3K4me2
H3K4me3	H3K4me3	H3K4me3
H3K79me2	H3K79me2	H3K79me2
H3K9ac	H3K9ac	H3K9ac
H4K20me1	H4K20me1	H4K20me1
Jund	Jund	Jund
Max	Max	Max
Maz	Maz	Maz
Mxi	Mxi	Mxi
Nfya	Nfya	Nfya
Nfyb	Nfyb	Nfyb
Nrf1	Nrf1	Nrf1
Nrsf	Nrsf	Nrsf
P300	P300	P300
Pol2	Pol2	Pol2
Pol2s2	Pol2s2	Pol2s2
Rad21	Rad21	Rad21
Rfx5	Rfx5	Rfx5
Smc3	Smc3	Smc3
Taf1	Taf1	Taf1
Tbp	Tbp	Tbp
Tr4	Tr4	Tr4
Usf2	Usf2	Usf2
Znf143	Znf143	Znf143
Znf274	Znf274	Znf274

3.2.2 Model Construction

We used gradient boosting (GB) trees to build the prediction model. GB is an ensemble method that combining weak learners like Decision Trees to a strong learner. GB fits new trees to minimize the loss or error, and it would assign more weights to observations that are hard to classify. Therefore, GB works great with categorical and numerical values and is suitable to be used on our datasets. We trained the model using ChIP-Seq, DNase-Seq, and ChIA-PET data on three cell lines for transcription and exon usage prediction, respectively, including GM12878, K562, and HeLaS3. The model parameters for each prediction task were determined using *sklearn.model_selection.GridSearchCV* with *cv=5* and *score='accuracy'* on corresponding datasets (Appendix A.1).

3.2.3 Validation Methods

We implemented three types of experiments for each cell line to assess the effectiveness of our approach. First, we adopted the stratified 10-fold cross-validation. The idea of the stratified helps to keep the same proportions of class labels in each fold. In each validation, one fold would be employed as validation set and the others folds would be adopted as training set. Such practice generally reflects the robustness of the model.

The second type of validation method is chromosome-split validation. We adopted the same chromosome-split strategy from the work of [104]. This work showed that there could be a large overlapping fraction when using functional-genomic data like ChIP-seq and DNase-seq data, and the authors proposed to use chromosome-split to exclude the similarity between training and test. Under this strategy, the samples from the same chromosome would be put into the same set, either training or test. The split of chromosome for training and test was listed in Table 3.6.

We also adopted the cross-cell validation to evaluate the model. We trained the model using all samples from one cell line and validated the model on all samples from different cell lines. We used the shared features when implementing cross-cell validation. Both the chromosome-split validation and the cross-cell validation enabled us to exclude the data similarity between samples in training and test datasets, hence could give us a fair implication about the robustness and effectiveness of the model.

Table 3.6: Chromatin in training and test dataset when implementing the chromosome-split validation.

	Training	Test
Chromosome name	chr1, chr2, chr4, chr6, chr7,	
	chr8, chr9, chr10, chr11,	chr3, chr5, chr13,
	chr12, chr14, chr15, chr17,	chr16, chr20, chr22
	chr18, chr19, chr21, chrX	

3.2.4 Evaluation Criteria

We used the criteria of Acc, Sn, Sp, AUROC, and AUPRC as defined in Chapter 2.4 to evaluate the model performances on different datasets in each classification task.

Due to the imbalanced datasets in exon usage prediction, we also calculated the random guess values for fair comparisons. The random guess values for AUROC and AUPRC were obtained as the average results by shuffling the true prediction scores 10 times, and the random guess value for accuracy was obtained as the average results by shuffling the true label 10 times.

Besides, we used XGBoost to obtain the feature importance, where the importance score for each feature was obtained as the average across all the decision trees within the GB model according to the amount that each attribute split point improves the performance measure. We applied function *xgboost.XGBClassifier* in Python using all data for each task to build the trees, then the feature importance scores were extracted using function *feature_importances_* from the XGBoost model for each dataset.

3.3 Results

3.3.1 Verified Transcripts can be Predicted from Transcription Factors and ChIA-PET Data

We first investigate if the verified transcripts can be predicted from transcription factors as well as ChIA-PET data. A verified transcript indicates that there is gene expression located at a *bona fide* gene locus as indicated by GENCODE hg19 gene annotation. We treat verified transcription as positive data while treating the unannotated transcripts as negative data. Different validation methods are evaluated on all three cell lines using the corresponding optimal parameters. The results are illustrated in Fig.3.2.

The accuracies of 10-fold validation range from 0.7638 to 0.7731, 0.7724 to 0.7791, and 0.7688 to 0.7787 for GM12878, HeLaS3, and K562 cell, respectively (Fig.3.2.a). The high accuracies confirm that transcription could be predicted from transcriptomics data with transcription factors and ChIA-PET. The prediction performance on the GM12878 cell is slightly lower than that of the other two cells. The average AUROC and AUPRC values for GM12878 cell of 10-fold validation are 0.8303 and 0.8436 separately, however, for HeLaS3 and K562 cells, they are 0.8354 and 0.8527, and 0.8364 and 0.8511, respectively. Notably, the small variations from 10-fold validation for all evaluation matrices on all cell lines, *i.e.* less than 0.0148, indicate model robustness.

Using the data from each cell line, we are able to identify some important features such as H3K4me3, H3K9ac, and H3K36me3 to distinguish the verified transcriptions from those not verified (Fig.3.3) in all three cell lines. The top identified important features show strong prediction powers. According to the average results of the 10-fold validation, training with the top 10 important features only slightly reduces 0.0047 to 0.0094 on the prediction accuracies and AUROC values from training with all features; training with the top 5 important features reduces 0.0083 to 0.0308 on the prediction accuracies and AUROC values from training with all features (Fig.3.4). Our findings of the important features are supported by several studies. First, as reviewed in Howe *et al.*, H3K4me3 may not be the initiator but the result of transcription, thus influencing processes such as splicing, transcription termination, the memory of previous states,

and transcriptional consistency [105]. Second, H3K9ac is reported to lead to increased transcription elongation via functioning in transcription to guide the Super-Elongation Complex (SEC) to chromatin and facilitate subsequent pol II pause release [106]. Third, H3K36me3 is thought to be a mark that is catalyzed by the SETD2 enzyme concomitantly with Pol2A transcriptional elongation [107]. Interestingly, we notice that H3K4me2 plays a more significant role in GM12878 and K562 cells than in HeLaS3 cells. Moreover, ChIA-PET Pol2A is also an important feature in K562 cells, and Pol2 is dominant in HeLaS3 but not very important in the other two cell lines.

We then evaluated our models on independent test datasets. The prediction results on chromosome-split validation (Fig.3.2.b) are similar to that of the 10-fold validation. The best accuracy is achieved on HeLaS3 data and the AUROC and AUPRC values for GM12878 cell are slightly worse than the other two cell lines. Interestingly, although the values of accuracy, AUROC, and AUPRC are comparable on all cross-cell validations (Fig.3.2.c), the model trained on HeLaS3 data leads to much lower sensitivity but much higher specificity values when testing on both GM12878 and K562 data. Conversely, the models trained on GM12878 and K562 data lead to higher sensitivity but lower specificity values when testing on HeLaS3 data. Such phenomenon further indicates the similarity of data characteristics between GM12878 and K562 cell line and their diversity from HeLaS3 cell as what we observed from the feature importance, which might be explained as the fact that the lymphoblastoid and myeloid cell lines are both suspension cells, while hela are adherent (monolayer) cells.

We also compare our GB model with other baseline models like Adaptive Boosting (AdaBoost), Random Forest (RF), Deep Neural Network (NN), and Convolutional Neural Network (CNN). The results of the average values for 10-fold validation and chromosome-split validation are shown in Fig.3.5 and Fig.3.6 separately. Although the average performance of the RF method on cross-validation is the highest for all three cells, the RF model only achieves slightly better performances than the AdaBoost model in the chromosome-split validation while performs worse than other models. GB model performs slightly worse than the RF model in 10-fold validation but it is the best model in chromosome-split validation, which shows the ability to predict independent data.

Fig. 3.2: The prediction results in verified transcript prediction. Sn (sensitivity) represents the positive prediction accuracy and Sp (specificity) represents the negative prediction accuracy. a. 10-fold validation. The boxplot here shows the spread and center of the 10-fold validation results. The five horizon lines from bottom to top represent the minimum (the smallest number in the data set), the first to the third quartile, and the maximum (the largest number in the data set), respectively. b. Chromosome-split validation. c. Cross-cell validation.

Fig. 3.3: Feature importance in verified transcript prediction.

Fig. 3.4: The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. Acc: Accuracy, AUROC: Area Under Receiver Operating Characteristic curve, AUPRC: Area Under Precision-Recall Curve, Sn: Sensitivity, Sp: specificity.

Fig. 3.5: The performance comparison among baseline models with the average values from the 10-fold validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. GB: Gradient Boosting Trees, AdaBoost: Adaptive Boosting (n_estimators=100), RF: Random Forest (n_estimators=100), CNN: Convolutional Neural Network (Conv1D(filters=40, kernel.size=2, strides=1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512), NN: Deep Neural Network (dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512).

Fig. 3.6: The performance comparison among baseline models from the chromosome-split validation in verified transcript prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: `n_estimators=100`, RF: `n_estimators=100`, CNN: Conv1D(40, 2, 1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512.

3.3.2 Transcription Abundance Prediction

TPM (Transcripts Per Million) is used to measure gene or transcript expression level in RNA-Seq. We aim to investigate if epigenomics can be used to predict expression level, and if it can leverage transcriptomics to measure and evaluate the accuracy of the predictions. We build a regression model to fit the TPM values. However, the performance with 10-fold validation is not good. The values of mean absolute error range from 3.77 to 4.32, 3.27 to 4.84, and 3.78 to 4.72, the root mean squared error range from 23.14 to 133.24, 19.54 to 183.83, and 21.30 to 133.02 for K562, HeLaS3, and GM12878, respectively. There are two possible reasons to explain the relatively poor regression prediction results. The first one is that the TPM values vary greatly, most samples have small TPM values and more than half of samples have TPM values equal to 0 (Fig.3.7). Another reason is that the features we used are mainly the counts of peaks from different studies. Lots of samples do not have peaks detected and thus contain lots of zero values, which could influence the model building.

Rather than using the regression, we classify the data into four classes according to their TPM values: `not_transcribed`, `lowly_transcribed`, `transcribed`, and `highly_transcribed`. More details can be found in 3.2.1. We validate the GB model via the 10-fold cross-validation. To avoid prediction bias caused by the imbalanced dataset, we down-sample the 3 classes which have more samples to the same size of the class which has the least number of samples during the training process.

The prediction results for 10-fold validation are illustrated in Fig.3.8.a. The average accuracies are 0.4240, 0.4516, and 0.4338 for GM12878, HeLaS3, and K562 cell lines, respectively. These values are higher when compared with random guess 0.25 in the balanced dataset, indicating that transcription abundance can be predicted using epigenomic data. Several HMs previously related to transcription are also found to be the most significant features by our model (Fig.3.9). H3K36me3, an HM-related to RNA Polymerase II passage and transcription fidelity maintenance [108], is found to be the most predictive across all three cell lines. This suggests that the levels of H3K36me3 are closely related to transcription activity. Besides, H3K79me2 is ranked the second most important feature except in K526 cells. H3K79me2 is a histone mark involved in alternative splicing regulation [109], indicating that the transcription machinery is closely

connected to exon splicing. Another transcription elongation-related HM, H3K9ac, is also identified as one of the most important features.

Interestingly, DNase Hypersensitive sites (a measure of open chromatin which is associated with an increased ability of the transcription machinery to access the region) and ChIA-PET loop anchors are also important in predicting transcription levels. We think that DNase may be an indicator of the open regions so that the transcription machinery can access the gene to produce transcripts. We reason that the presence of ChIA-PET loop anchors may reflect that the gene is interacting with distal regulatory elements and these regulatory elements may harbor transcription factor binding motifs that can recruit TFs or mediators, which together orchestrate a stable transcription structure for RNA Polymerase II [110].

The importances of H3K36me3, H3K79me2, and H3K9ac in predicting gene expression have been also shown by other studies. Rosa *et al.* demonstrated that H3K9ac is one of the most important HMs in identifying the expression of promoters with different CpG content in human T-cells [111]. Besides, Dong *et al.* showed that H3K36me3, H3K79me3, and H3K9ac are predictive of gene expression from either CAGE or RNA-Seq data [112]. Our results are consistent with these reported correlations between gene expression and HMs. Taken together, we demonstrate that specific HMs, chromatin loop anchors, and open chromatin are predictive of transcription abundance.

Again, the top identified important features show strong prediction powers in transcription abundance prediction (Fig.3.8.b). But the RF performs best on this classification problem (Fig.3.8.c-d).

3.3.3 Exon Usage Coefficient Prediction

We perform exon usage prediction using the exon usage coefficient as a label. The exon usage coefficient indicates the number of reads at a particular exon present in a particular cell. We treat the samples whose exon usage coefficient values are non-Nan (the exon is included in one or more transcripts) as positive while treating those with Nan (the exon is not included in any transcripts) value as negative. Because we consider all exons, and many exons do not have reads, there are much more negative samples and hence

Fig. 3.7: Histogram plot for TPM values of three cell lines. From top to bottom: K562, HeLaS3, and GM12878. Left: TPM values $\leq Q3$, right: TPM values $> Q3$.

Fig. 3.8: Results for transcription abundance prediction. a. The prediction results of 10-fold validation. The bottom table represents average accuracy. b. The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation for GM12878, HeLaS3, and K562 cell. Acc: Accuracy, AUROC: Area Under Receiver Operating Characteristic curve, AUPRC: Area Under Precision-Recall Curve, Sn: Sensitivity, Sp: specificity. c. The performance comparison among baseline models with the average values from the 10-fold validation for GM12878, HeLaS3, and K562 cell. AdaBoost: n_estimators=100, RF: n_estimators=100, CNN: Conv1D(40, 2, 1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512. d. The performance comparison among baseline models (parameters are same as above) from the chromosome-split validation for GM12878, HeLaS3, and K562 cell.

Fig. 3.9: Feature importance in transcript abundance prediction.

the dataset is imbalanced. We evaluate the model performances mainly based on the criteria of accuracy, AUROC, and AUPRC. With large size of the negative dataset, using AUPRC obtained from the precision ($TP/(TP + FP)$) and recall ($TP/(TP + FN)$) makes it possible to evaluate the prediction of a classifier on the minority class by excluding the influence of TN.

In the 10-fold validation prediction (Fig.3.10.a), the average accuracy values for GM12878, HeLaS3, and K562 cells are as high as 0.7700, 0.7893, and 0.8014, respectively. The same tendencies also can be found from the corresponding AUROC values for GM12878, HeLaS3, and K562 cells, which have average AUROC values of 0.6825, 0.6979, and 0.7001, respectively.

H3K36me3 is found to be important in distinguishing exon usage levels in all three cell lines (Fig.3.11), which are similar to the situation of verified transcription classification. H3K27ac is another important feature for the exon usage prediction, but it is not a very important feature in transcription prediction. The observation that H3K27ac is associated with gene exon usage has not been made before, and we suggest that perhaps

different enhancer usage (since enhancers associated with H3K27ac) may be involved in specifying which exons are to be transcribed and included in the mRNA by splicing. Also, Pol2A ChIA-PET chromatin interaction anchors and H3K4me2 are important features for exon prediction in GM12878 and K562 cells, and H3K9ac is an important feature for exon prediction in GM12878 and HeLaS3 cells. Similarly, the top identified important features show strong prediction powers in exon usage prediction. According to the average results of the 10-fold validation, training with the first 10 important features only reduces 0.0126 to 0.0569 on the prediction accuracies, AUROC and AUPRC values from training with all features; training with the first 5 important features reduces 0.0169 to 0.0929 on the prediction accuracies, AUROC and AUPRC values from training with all features (Fig.3.12).

The performances of chromosome-split validation (Fig.3.10.b) and cross-cell validation (Fig.3.10.c) are similar to that of the 10-fold validation. The prediction results on K562 cell data have higher accuracy values but lower AUPRC values considering the data amount ratio. However, compared to the random prediction results, the models work effectively and robustly in predicting the exon usage on independent test datasets. We show some exons that have predicted exon usage in GM12878 but not HeLaS3 cell (Fig.3.13) and in HeLaS3 but not GM12878 cell (Fig.3.14) trained with K562 cell. The RNA-Seq signal of H1hESC, K562, GM12878, and HeLaS3 cells are shown in RPKM (Reads Per Kilobase of transcript, per Million mapped reads) to represent the normalized transcript expression. The results here illustrate that our prediction is in agreement with the real detected exon usage.

We also compare our GB model with other baseline models (AdaBoost, RF, CNN, and NN) on the average values for 10-fold validation and chromosome-split validation (Fig.3.15 and Fig.3.16). Results demonstrate that the GB method is the second best on 10-fold cross-validation and the best on independent test datasets (chromosome-split validation).

3.4 Discussion

Exon usage is important in revealing the abundance of different transcript variants from the same gene. Pre-mRNA alternative splicing can result from different transcription

Fig. 3.10: The prediction results in exon usage prediction. Sn (sensitivity) represents the positive prediction accuracy and Sp (specificity) represents the negative prediction accuracy. a. 10-fold validation. The boxplot here shows the spread and center of the 10-fold validation results. The bottom table represents the random values. b. Chromosome-split validation. The black horizon lines represent the random results. c. Cross-cell validation. The black horizon lines represent the random results.

Fig. 3.11: Feature importance in exon usage prediction.

Fig. 3.12: The performance comparison among training with all features, training with the first 10 important features, and training with the first 5 important features with the average values from the 10-fold validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell.

Fig. 3.13: Example of exon that has predicted exon usage in GM12878 but not HeLaS3 cell. RNA-Seq signal of H1hESC, K562, GM12878, and HeLaS3 cells are shown in RPKM. Example exons are highlighted in light blue. The example exons in this figure are those predicted as positive in GM12878 while negative in HeLaS3 cell using K562 model. For each example, a total RNA-Seq signal across the exon region is calculated and plotted in barplot. We can see that in this category, the total RNA-Seq signal of GM12878 is higher than HeLaS3. The examples showed here are: a. ENSE00003890465.1; b. ENSE00000767123.1; c. ENSE00003836981.1.

Fig. 3.14: Example of exon that has predicted exon usage in HeLaS3 but not GM12878 cell. RNA-Seq signal of H1hESC, K562, GM12878, and HeLaS3 cells are shown in RPKM. Example exons are highlighted in light blue. The example exons in this figure are those predicted as positive in HeLaS3 while negative in GM12878 cell using K562 model. For each example, a total RNA-Seq signal across the exon region is calculated and plotted in barplot. We can see that in this category, the total RNA-Seq signal of HeLaS3 is higher than GM12878. The examples showed here are: a. ENSE00001904186.1; b. ENSE00003508391.1 and ENSE00000735744.1; c. ENSE00000801685.4.

Fig. 3.15: The performance comparison among baseline models with the average values from the 10-fold validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: `n_estimators=100`, RF: `n_estimators=100`, CNN: `Conv1D(40, 2, 1, activation='relu')`, dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', `optimizer='adam'`, `epoch=50`, `batch size=512`, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', `optimizer='adam'`, `epoch=50`, `batch size=512`.

Fig. 3.16: The performance comparison among baseline models from chromosome-split validation in exon usage coefficient prediction for GM12878, HeLaS3, and K562 cell. AdaBoost: `n_estimators=100`, RF: `n_estimators=100`, CNN: Conv1D(40, 2, 1, activation='relu'), dropout 20%, dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512, NN: dense to 400 with 'relu', dropout 20%, dense to 2 with 'relu', optimizer='adam', epoch=50, batch size=512.

kinetics, often changing transcript abundance and biological function, which can be implicated in diseases [113]. Here we predicted exon usage using functional genomic data for the first time. Moreover, we extended previous transcription prediction approaches by including chromatin loop data (*i.e.* ChIA-PET data) in the prediction. Our prediction results on 10-fold validation as well as the chromosome-split and cross-cell validation which ensure the independence of the data illustrated the effectiveness and robustness of our approaches. The different validation methods together with the large datasets on three cell lines also indicated that our model is scalable.

Specifically, in the transcription prediction part, the machine learning framework revealed close relationships between transcription abundance and transcription-related HMs, including H3K36me3, H3K79me3, and H3K9ac. These important features again re-emphasized the importance of epigenetic marks in maintaining a permissive and high-fidelity environment for gene transcription. Moreover, our gradient boosting tree model was able to recapture these features, as they were also reported to be important in predicting gene expression using conventional regression models [111, 112].

HMs are engaged in differential exon usage regulation in human cells [114]. For example, H3K9ac hyperacetylation at a splice site of the NCAM gene can result in skipping of an exon in neuron cells [115]. H3K36me3 histone marks can be recognized and bound by splicing regulators (such as MRG15 and Psip1) and thus regulating the splicing of the transcript [114]. Importantly, our model also identified H3K36me3 and H3K9ac as important features in predicting exon usage. The link between splicing and chromatin interactions is unclear. While it has been shown previously that chromatin loops can bring exons into close spatial proximity with promoters and enhancers [102], it was unclear how important chromatin loops were in predicting splicing occurrences. Here, we observed that RNA Pol2A ChIA-PET anchors are important in predicting transcription and exon usage.

RNA polymerase II produces a pre-mRNA during transcription [116] and a large number of nuclear localized lncRNAs are transcribed by RNA polymerase II [117]. Splicing mainly occurs while transcription is ongoing, and *in vivo* studies have shown that the spliceosomes are physically close to RNA Polymerase II [118, 119]. Splicing will give rise to different exon usage because different exons may be included in the final transcripts.

Our model identified H3K36me3, H3K9ac, and RNA Pol2A ChIA-PET anchors as important for transcription prediction as well as exon usage prediction, which highlights the tight connection between splicing and transcriptional machinery. Based on our results, we extended this "splicing-transcription" model whereby we proposed that the splicing regulators, RNA polymerase II, and chromatin loop anchors bound by RNA polymerase II all come together in close proximity. This may be promoted in part by key HMs such as H3K36me3 and H3K9ac helping to recruit splicing regulatory proteins (Fig.3.17).

We speculated that perhaps chromatin interactions between the exon being transcribed and distal regulatory elements may help to stabilize the transcription machinery [110]. When the splicing and/or transcription machinery is assembled at the transcript, it creates structural scaffolds for DNA or RNA to fold and interact with each other. Such interactions between DNA/RNA and distal regulatory elements can be captured by chromatin interaction profiling techniques like ChIA-PET. Therefore, this interaction information was learned by our model when predicting transcript and exon usage.

Taken together, the data we collected from different sources which represent different biological processes such as interaction with DNA, regulation, and chromatin interaction show some relativity with the RNA transcription. Specifically, our prediction results showed that transcription-related HMs are important in predicting transcript abundance and exon usage in human cells. More importantly, we showed that chromatin interaction data were also important in predicting transcript and exon usage, suggesting a close relationship between transcription, splicing, and chromatin structural organization.

3.5 Chapter Summary

In this chapter, we successfully identify the transcript from DNA. We construct GB models to infer the transcript and exon usage using transcription factors and chromatin-related data from different cell lines, where the GB models perform better than other baseline methods in most cases. Our results show that transcription-related HMs are important in predicting transcript abundance and exon usage in human cells. More importantly, we show that chromatin interaction data are also important in predicting

Fig. 3.17: Schematic of our proposed model of alternative splicing, which gives rise to differential exon usage, chromatin loops and transcription machinery. In this model, we speculate that HMs such as H3K36me3, H3K9ac, and chromatin interactions, can bind to splicing regulators and transcription machinery. This would explain our findings that these marks predict high exon usage.

exon usage, suggesting a close relationship between transcription and chromatin structural organization. In the next chapter, we will focus on distinguishing different types of transcripts to annotate the long non-coding RNAs.

Chapter 4

RNA Coding Potential Prediction with Nucleotide Bias Information¹

4.1 Introduction

Long non-coding RNAs (lncRNAs) comprise a majority of the transcriptome [121]. The advances in deep sequencing technologies and the fulfillment of diverse genome projects reveal significant functions of lncRNAs, which make them being regarded as treasures rather than the "dark matter" as previously thought [122]. lncRNAs are involved in a great variety of biological processes, including transcriptional regulation, chromatin modification, chromatin remodeling, and gene expression [123–126]. They have also been suggested to relate to diseases like diabetes and cancer [16, 17].

Correctly identifying the lncRNAs from other transcripts is the prerequisite to explore lncRNAs and their functions, hence accurate and rapid computational approaches are needed for newly annotated transcripts. Plenty of computational approaches have been proposed to identify lncRNAs from coding RNAs (mRNAs). These approaches mainly can be classified into two groups according to if the alignment is needed: alignment-based approaches and alignment-free approaches. Alignment-based approaches depend on the existing databases. For example, Kong *et al.* developed Coding Potential Calculator

¹The work in this chapter has been published in [120].

(CPC), where three of its features were calculated according to a known database of protein sequence [50]. Similarly, Lin *et al.* calculated the likelihood ratio by comparing 2 well-constructed models corresponding to coding and non-coding genes separately and proposed PhyloCSF [51]. However, relying heavily on known databases is an apparent limitation for the above approaches. These methods may perform poorly under the circumstance when there are broad changes on the newly identified test samples.

To go beyond the bounds of the alignment-based approaches, alignment-free approaches are developed. The alignment-free approaches are more flexible than alignment-based ones. They extract the intrinsic information from the given sequences directly. One of the most common features to be used to represent the sequence intrinsic information is the longest open reading frame (ORF). ORF refers to the DNA segments that can be encoded to proteins. The ORFs of mRNAs usually are longer than that of non-coding ones, as the probability to formulate a putative ORF to translate is extremely small. Therefore, the length of the longest ORF is popular and has been proved to be work efficiently in the classification between mRNA and lncRNA. Another popular feature is k-mer. k-mer feature reflects the internal features of the sequence by counting the frequencies of nucleotides. Some prediction tools have been developed based on the longest ORF and k-mer frequencies. For example, CPAT was constructed with a logistic regression model using the longest ORF-related features along with Fickett and hexamer features [52]. And tools like CNCI, PLEK, DeepLNC, and Hugo’s SVM method used k-mer features with relatively small values of k , *i.e.* $k \leq 6$, which were reported to perform effectively to distinguish lncRNAs from mRNAs [53, 54, 60, 63]. Some authors believed that a large k value in k-mer can suggest more repeats or spatial information specific to lncRNAs, therefore, tools like FEELnc were developed by adopting a long k-mer with $k = 12$ [59]. Other features representation methods such as one-hot vector and high-level features learned by the network also have been successfully applied to this classification problem [61, 64, 65]. These alignment-free approaches are reported to achieve overall better performances than those alignment-based methods.

However, not too much attention has been paid to RNA transcripts with small ORFs (sORFs). sORFs are easy to be regarded as non-coding. Except for the short length, other possible reasons for redeeming sORFs as non-coding consist of i) lacking the experimental corroboration for their functions, and ii) the annotation problem brought about

by their vast amount [127]. The mRNAs with sORFs may fail to be classified as coding ones due to the wrong initial assumptions. The work of mRNN [65] and CPPred [62] tested their models on sORF type data, but the results were far less desirable compared to normal type data. For example, only 63.34% *human* samples of mRNAs with sORFs and 46.81% such samples in *mouse* species were correctly predicted by CPPred on its test datasets separately. There are still possibilities for the improvement on the prediction of sORF type mRNAs.

In this chapter, we expect to construct models to accurately predict the coding potential of RNA transcripts, especially for those with sORFs. We extract the information of the nucleotide preference around the start codon to a feature named nucleotide bias. This feature is illustrated to work effectively on RNAs with sORFs. We also demonstrate that k-mer features with some gaps within the components could play more dominant roles than that without gaps in this classification problem. Besides, we adopt a wide range of feature representation approaches including the length and coverage for the longest ORF, Fickett score, and hexamer score. We rank the features and determine the different feature subsets for normal type data and sORF type data using the minimum Distribution Similarity (mDS) algorithm [128]. We construct DeepCPP - a deep neural network for RNA coding potential prediction, and evaluate it on samples from multiple species with both the normal and sORF type transcripts. We also compare it with 7 newly published state-of-the-art approaches. DeepCPP delivers prediction accuracies for normal type data with values of 0.9820, 0.9578, and 0.9616, and it improves upon other approaches for sORF type data with at least 1.08%, 17.09%, and 3.14% regarding the accuracies on our newly constructed *human*, vertebrate, and insect test datasets, respectively.

4.2 Materials and Methods

4.2.1 Dataset

We downloaded the same training data as the work of CPPred to train the *human* model [62]. The training data for two integrated models were newly collected. As to test

data, we adopted 4 previous test datasets from [62], and further constructed 6 new test datasets for three kinds of species, where each species comprises a normal and an sORF type test dataset.

We downloaded the training and test datasets of *human*, and test datasets of *mouse* from the work of [62]. Its *human* coding RNAs were collected from NCBI RefSeq [39] with transcript status ‘KNOWN’ and ncRNAs were from Ensembl release 90 [44]. Three sequences containing the letter ‘N’ were removed. The training and test datasets in CPPred of the *human* were randomly split with ratio 2:1, and the *human* test dataset was kept as sequences with no less than 80% identity with training dataset using CD-HIT [129]. The data for its *mouse* dataset was downloaded from Ensembl. sORF type RNA test datasets were constructed from the original *human* and *mouse* test data separately, where the RNAs are those whose ORFs smaller than 303 nucleotides in length.

Besides the above datasets, we further constructed a new *human* test dataset as well as its sORF type dataset. The mRNA data in our new test dataset was selected as the newest released sequences in NCBI from 18th May 2018 onward with search limitations: “*Homo sapiens*”[*Organism*] AND *srcdb_refseq_known*[*prop*] AND *biomol_mrna*[*prop*]. And the ncRNAs were collected from Ensembl release 97 (ftp://ftp.ensembl.org/pub/release-97/fasta/homo_sapiens/ncrna/). Three types of sequences were removed from the original downloaded data: 1) overlapped sequences with Ensembl release 90, 2) sequences with letters other than ‘ACGT’, and 3) ncRNAs with lengths less than or equal to 200-nt. Finally, our final new *human* test dataset comprises 3,373 mRNAs and 23,915 ncRNAs, it should be noticed that all 23,915 ncRNAs are lncRNAs.

The mRNA samples for the new sORF type RNA dataset were selected as those whose ORF lengths are less than or equal to 150 codons. Some studies reported that an ORF should contain at least 100 codons [130] or 150 codons [131]. Considering that the number of mRNA samples whose ORF no longer than 100 codons in our new test datasets was quite small, *i.e.* 48, we decided to use 150 codons as the cut-off. 232 mRNAs were chosen in this way, and 232 lncRNAs were selected randomly in the same way to construct the new sORF RNA dataset.

We also built two integrated models from multiple organisms, one for vertebrates and the other for insects. We used species of *mouse* and *zebrafish*, and *fruit fly* (*Drosophila*

melanogaster) and *mosquito* as the training species separately for two integrated models. Their sequences for mRNA were downloaded from NCBI, which were sorted by release date and following the selecting rules of "*species*"[*Organism*] AND *srcdb_refseq_known*[*prop*] AND *biomol_mrna*[*prop*]. The sequences for *mosquito* mRNA were from species of *Anopheles gambiae* and *Aedes aegypti*, which are important disease transmission vectors. The corresponding sequences of ncRNA for these species were downloaded from Ensembl release 97 except *mosquito*, as there were no such data in the Ensembl database. As an alternative, the ncRNAs of *mosquito* were downloaded from NCBI with selecting rules "*Culicidae*"[*Organism*] AND (*ncrna*[*filter*] OR *biomol_rrna*[*PROP*] OR *biomol_trna*[*PROP*]). Once again, the sequences with letters other than 'ACGT' and ncRNAs whose length less than or equal to 200 were removed from our datasets.

The training and test data for mRNAs of the above species and ncRNAs of *mosquito* were split with a ratio of 8:2 according to the release date, the first 80% old samples were used as training and the 20% relatively new samples were adopted as test. The split of ncRNA samples to training and test for species of *mouse*, *zebrafish*, and *fruit fly* followed a different way due to the lack of time information in Ensembl database. For these samples, those overlapped with Ensembl release 92 for vertebrate species and release 90 for *fruit fly* were adopted as training data, and the other samples were regarded as test samples. The reason why we chose the release 92 for vertebrate species is that we want to put more data into training due to the relatively small amount of ncRNA data in these species compared to their mRNA samples. And we chose release 90 for *fruit fly* because its release 92 contains too little non-overlapping data with release 97, *i.e.* only 48 such samples. Additionally, as the amount of mRNA was much larger than that of ncRNA in some cases, to improve the robustness of the models as well as keep the data balance, we used CD-HIT [129] with a threshold of 0.8 to exclude the redundancy for mRNA training data whose amount exceeding 10,000 in that one species. And the final training dataset for each taxonomic group was obtained by integrating the training data for species belonging to that taxonomic group.

To balance the amount of mRNA and ncRNA test samples as well as validating the performance of our models in a more challenging way, *e.g.* cross-species validation, we downloaded samples from other species. We collected *pig* & *dog* mRNAs, *honeybee*

Fig. 4.1: The description of data amount, dataset annotation, and the procedure to build datasets. The number in each text box is the data amount left after filtering in the last step. The color of the text box is to distinguish ncRNA and mRNA, vertebrate and insect.

(*Apis Mellifera*) mRNAs, and *honeybee* ncRNAs from NCBI following the same filter criteria as above, and downloaded ncRNA data of *pig & dog* from Ensembl release 97. After filtering, 5,699 mRNAs and 9,361 ncRNAs for *pig & dog*, and 3,742 mRNAs and 3,148 ncRNAs for *honeybee* were obtained respectively. The final test datasets for two integrated models were constructed by integrating these data with the split test data above. Similarly, we also constructed sORF type RNA test datasets for each integrated model from the original test datasets by selecting mRNAs whose ORFs are 100 codons or shorter in length, and the same amount of ncRNAs with comparable ORF lengths.

The flow to construct each new dataset was demonstrated in Fig.4.1. The information for all datasets used in this work was shown in Table 4.1.

4.2.2 Feature Extraction

We adopted a wide range of approaches to extract intrinsic information from raw sequences, including the longest ORF-related features [52], sequence composition features such as Fickett score [132], Mean hexamer score [52, 62, 66], k-mer [63] and g-gap, and the codon preferences around start codon represented by our newly proposed Nucleotide bias feature.

Table 4.1: A summary of the datasets used in this work.

Dataset name	Annotation	Number of ncRNAs	Number of mRNAs
Human training dataset from CPPred	D_{train_H}	24,162	33,359
Human test dataset from CPPred	D_{test_H}	8,240	8,557
Human sORF RNA test dataset from CPPred	$D_{test_H_sORF}$	641	641
Mouse test dataset from CPPred	D_{test_M}	19,930	31,100
Mouse sORF RNA test dataset from CPPred	$D_{test_M_sORF}$	1,000	846
New human test dataset	$D_{test_H_new}$	23,915	3,373
New human sORF RNA test dataset	$D_{test_H_newsORF}$	232	232
Vertebrate training dataset	D_{train_V}	18,635	31,706
Vertebrate test dataset	D_{test_V}	12,055	16,045
Vertebrate sORF RNA test dataset	$D_{test_V_sORF}$	588	588
Insect training dataset	D_{train_I}	12,465	25,181
Insect test dataset	D_{test_I}	5,821	12,734
Insect sORF RNA test dataset	$D_{test_I_sORF}$	1,529	1,529

4.2.2.1 The Maximum ORF Length and Coverage

The ORF denotes a region of nucleotides with the potential ability to be translated to protein, where the transcription initiation is indicated by the ATG codon (start codon) and ended at the TAG, TAA, or TGA codon (stop codon). As not all sequences in our ncRNA dataset have complete ORFs, we treated the data as two types: sequences with complete ORFs and sequences with incomplete ORFs. We defined the complete ORF as the sequence with both start and stop codon. And we defined the incomplete ORF as the sequence that missing a start or stop codon if there is no complete ORF in a sequence. For example, a sequence begins with the ATG codon and stops at the last codon of its sequence, or a sequence starts from the first codon of its sequence and stops at a stop codon. The maximum ORF length represents the number of nucleotides for the longest ORF or incomplete ORF, denoted as l_{ORFmax} . A 2-dim vector $[c_1, c_2]$ was used to represent the ORF coverage and the sequence type, where c_1 represents the coverage of complete ORF and c_2 represents the coverage of incomplete ORF. The ORF coverage is calculated as dividing l_{ORFmax} by the length of the whole sequence.

4.2.2.2 Hexamer Score

The hexamer score represents a relative hexamer bias of the longest ORF. To calculate it, firstly, the 6-mer frequency of the longest ORF for each sequence was counted. Then by averaging 6-mer frequencies in positive and negative training data separately, the mean values of 6-mer vectors for in-frame sequences can be obtained, denoted by $M_c(h_i)$ for the coding model and $M_{nc}(h_i)$ for the noncoding model, where $i = 1, 2, \dots, 4096$ represents the 6-mer components. Therefore, given a hexamer sequence $S = h_1 h_2 \dots h_n$, its mean hexamer score is obtained as the log-likelihood ratio between $M_c(h_i)$ and $M_{nc}(h_i)$:

$$hexamerscore = \frac{1}{n} \sum_{i=1}^n \log \frac{M_c(h_i)}{M_{nc}(h_i)} \quad (4.1)$$

4.2.2.3 Fickett Score

Fickett score reflects the combined effect of codon usage bias and nucleotide composition in the whole sequence, which is different from the hexamer score that only depending on

the longest ORF region. Composition and position frequencies were calculated for four bases: A, C, G, and T, and then they were transformed to a single value using a predefined lookup table in Fickett's work [132]. However, in this work, rather than using the 1-dim score, we chose to use the original nucleotide position frequency, which represents the nucleotide preference in a codon. We used a four-dimension vector $[f_A, f_C, f_G, f_T]$ to denote Fickett position feature. Suppose

$X_1 = \text{Number of } X \text{ in the first position in a codon}$

$X_2 = \text{Number of } X \text{ in the second position in a codon}$

$X_3 = \text{Number of } X \text{ in the third position in a codon}$

Fickett position feature is calculated as

$$f_X = \frac{\max(X_1, X_2, X_3)}{\min(X_1, X_2, X_3) + 1}, X \in A, C, G, T \quad (4.2)$$

4.2.2.4 k-mer, g-gap, and g-bigap

k-mer nucleotide composition is one of the most commonly used approaches to convert the raw sequences to vectors. It counts the frequencies for its 4^k types of elements according to the following formula:

$$kmer(i) = \frac{\text{The number for each } k \text{ neighboring nucleic acid component } (i)}{n - k + 1} \quad (4.3)$$

where $kmer(i)$ refers to the frequency of the i th k neighboring nucleic acid component from its 4^k possibilities, which is obtained by dividing the counts for one type composition by the total counts corresponding to k . Similarly, g-gap considers the composition of two nucleic acids with g gaps and g-bigap represents the composition of two dinucleotides with g gaps. In this work, we considered 1-mer, in-frame 3-mer (3-mer for longest ORF), and the frequencies of two dinucleotides with g gap with $g = 2, 3$.

4.2.2.5 Nucleotide Bias Feature

Start codon suggests the beginning of translation, but not vice versa. Pieces of evidence show that the nucleotides around the start codon play some roles in regulating the initiation of translation [133–136]. Nakagawa *et al.* found that the first codon before and after the start codon in true coding sequences (CDS) showed noticeable bias when they applying G-statistic to evaluate the nucleotide appearance on 10,012 *human* genes [133]. Their findings of the nucleotide preferences that may promote the initiation of translation were also reported by other researches [134–136]. Nakagawa *et al.* also demonstrated that the species within the same taxonomic group of eukaryotes tend to have similar nucleotide patterns, where the G-values for the nucleotides from the first three codons before the start codon and the first codon after the start codon are relatively high, especially for the first codon before and after the start codon.

Based on the evidence discussed above, we presented an original feature extraction approach named nucleotide bias to classify ncRNAs and mRNAs. The nucleotide bias feature demonstrates the nucleotide preference surrounding the start codon. If we use +1, +2 and +3 to represent the nucleotide position of the start codon, the nucleotide position of the codon before and after the start codon would be represented as -3, -2 and -1, and +4, +5 and +6, and so on. We obtained the prior nucleotide preferences from training dataset by averaging the frequency of nucleotide at interested locus around the start codon for the longest ORF. We used $p_{mRNA(x_i)}$ and $p_{ncRNA(x_i)}$ to denote the corresponding frequency of nucleotide x at position i for coding and non-coding samples. The value of nucleotide bias feature for each sample is obtained by summing the log ratios between $p_{mRNA(x_i)}$ and $p_{ncRNA(x_i)}$ of the corresponding x at position i :

$$nucleotide \quad bias = \sum_{i \in C} \log \frac{p_{mRNA(x_i)}}{p_{ncRNA(x_i)}}, x \in \{A, C, G, T\}, C = \{-3, -2, -1, 4, 5, 6\} \quad (4.4)$$

where C represents the set of position regarding the start codon of the longest ORF with elements of -3, -2, -1, 4, 5, and 6.

4.2.3 Model Construction

We built DeepCPP to assess the RNA coding potential and make the prediction for each input RNA to be coding or non-coding. Our program was constructed with 2 Convolutional Neural Network (CNN) layers, as the CNN model achieved better performances compared to other machine learning models (Appendix B.1). Considering the potential differences of characteristics for normal type data and sORF type data, we constructed different models for them. Both models used the same training data in each species but different features. We selected different feature subsets by applying mDS feature ranking method on *human* training dataset and *human* sORF test dataset, where the corresponding feature subsets were determined according to the best accuracy achieved with the first n features ranked by mDS using a CNN model. The architectures and parameters for normal type and sORF type model can be found in Appendix B.2. We used the same feature subsets determined by *human* data to train the vertebrate and insect integrated model.

Because the parameters in each model were randomly initialized, we trained three models and ensembled them into one. The output probability score of DeepCPP was taken as the average values from three models, the final predicted label was determined as coding and non-coding with a cut-off value of 0.5 with the normalized output from each model.

The working mechanism of DeepCPP was shown in Fig.4.2. The features for each input sequence were extracted and the corresponding feature subset was determined. The feature vectors then went through the trained models according to the choice of normal model or sORF model and the choice of species. Finally, the predicted probability score would be output and a label of coding or non-coding would be assigned for each input sample.

4.2.4 Evaluation Criteria

We used a wide range of criteria as defined in Chapter 2.4 to evaluate our model as well as other approaches, including Acc, Sn, Sp, F-score, Hm of Sn and Sp, and Mcc.

Fig. 4.2: The workflow of DeepCPP. a. Overall structure. b. Feature extraction approaches and the sizes of each feature subset. c. Convolutional Neural Network structure, including two CNN layers, a flatten layer and a dropout layer with dropout rate of 20%, and two dense layers.

Additionally, the McNemar test was employed to compare two models from a statistic respective, whose chi-square was calculated in a continuity correction version as:

$$\chi^2 = \frac{(|B - C| - 1)^2}{B + C} \quad (4.5)$$

where B denotes the samples that are correctly predicted by Model I but wrongly predicted by Model II, and C denotes the samples that are correctly predicted by Model II but wrongly predicted by Model I. The corresponding p-value was obtained according to the cumulative distribution function (CDF) of chi-square distribution using python.

4.3 Results and Discussion

4.3.1 Feature Exploration

We count the frequencies of nucleotides at the position of -3, -2, -1, +4, +5, and +6 around the start codon for the longest ORF from our training data to see if we could observe the same tendencies as that reported by previous works. We demonstrate the results using sequence logo for species of *human*, vertebrate, and insect (Fig.4.3.a-c). We observe that the nucleotides with the largest frequencies in our mRNA training data of *human*, vertebrate, and insect are *ACCGCG*, *ACCGCG*, and *AAAGCG*, respectively. The results here are consistent with what had been reported in the work of [133], for example, *A/G* are favored by position -3 and *C* is favored by position +5 in *human*. By contrast, the frequencies of nucleotides on these positions are more evenly distributed in ncRNA training data, and they do not show similar patterns as that of mRNA. This phenomenon indicates that the nucleotide bias feature which measuring the bias of nucleotides around the start codon may contribute to the ncRNA and mRNA classification.

We then evaluate the performance of the nucleotide bias feature on our data and compare it with other commonly used features, including the maximum ORF length and coverage, 1-mer, mean hexamer score, and Fickett score. All the features to be compared have been suggested to work effectively in this classification problem [52, 62, 66, 67]. We train a CNN model using different features and demonstrate the prediction performances on two previously used test datasets, where one is normal type (D_{test_H}) and the other is sORF type ($D_{test_H_sORF}$) (Fig.4.3.d-e). The average values for five times running are shown.

The maximum ORF length feature performs dominantly on D_{test_H} but poorly on $D_{test_H_sORF}$, it achieves values of more than 0.9 but less than 0.2 regarding the F-score and Hm on two test datasets separately. Because the ORF for coding RNA is usually longer than that of the non-coding ones based on the truth that the probability to form a long putative ORF in noncoding sequences is extremely small. Therefore, the maximum ORF length feature cannot distinguish the coding and non-coding RNAs when their ORFs are in similar lengths. Conversely, nucleotide bias feature performs less powerful on D_{test_H} but more noticeable on $D_{test_H_sORF}$. Another useful feature on sORF type

data is the hexamer score. The feature of nucleotide bias and hexamer score achieve the largest two values from all measurements on $D_{test_H_sORF}$. Their values of accuracy and F-score are over 0.6 while the accuracy and F-score values for other approaches are all below 0.6. Specifically, the nucleotide bias feature realizes higher values of Hm value than that of the hexamer score feature on $D_{test_H_sORF}$. The values of specificity and sensitivity for nucleotide bias are 0.5382 and 0.6817, and for hexamer score feature they are 0.4540 and 0.7863, respectively. The gap between its specificity and sensitivity from the nucleotide bias feature is smaller than that of the hexamer score feature. Therefore, another advantage of the nucleotide bias feature is that it performs more evenly on different classes without too much bias.

k-mer is a type of feature that is frequently used to distinguish coding and non-coding RNAs [53, 54, 60, 63]. In this work, we adopt $k = 1$ and $k = 3$ to represent the whole sequences and the longest ORF sequences. 3-mer feature in ORF sequences could reflect some information related to translation, as 1 amino acid is composed of 3 nucleotide acids. We evaluate other choices of k ($k = 2, 4$) and the cases of g-gap and g-bigap ($g = 1 - 3$) in revealing the coding potential of transcripts. The corresponding ROC and PRC curves on two types of data (D_{test_H} and $D_{test_H_sORF}$) are shown in Fig.4.3.f-g separately.

In both two test datasets, the best performances are achieved by the 2-bigap feature sets and the second best performances are achieved by the 3-bigap feature sets. Compared to the continuous 4-mer features, the g-bigap features show much better performances. For example, the g-bigap features improve at least 6.64% and 7.32% on D_{test_H} regarding the Area Under the ROC and PRC, and at least 9.04% and 8.11% on $D_{test_H_sORF}$ compared to the continuous 4-mer features, respectively. Our results suggest that g-bigap feature, especially $g = 2, 3$, could play more dominant roles in assessing the RNA coding potential. But these features were rather ignored by the previous approaches which mainly addressed the continuous k-mer feature [53, 54, 59, 60, 63].

4.3.2 Feature Selection

The size of our original feature set is 589, where the feature types and their numbers can be found from Fig.4.2.b. However, maybe not all of these features could contribute to

Fig. 4.3: Evaluation of new features used in this work. a-c. Sequence logo representation of the training dataset for a. *human* species, b. *vertebrate* species, and c. *insect* species. The first row and the second row are corresponding to lncRNA and mRNA separately. d-e. Performance of each feature set on d. CPPred *human* test dataset D_{test-H} and e. CPPred *human* sORF test dataset $D_{test-H-sORF}$. f-g. ROC (left) and PRC curve (right) for continuous or discontinuous k-mer feature sets on f. D_{test-H} and g. $D_{test-H-sORF}$.

the classification. Feature selection can be employed to remove the redundant features to enhance the training efficiency.

We adopt mDS method to rank the features according to their abilities to separate different classes. mDS method uses the relative entropy to measure the similarity between the distributions in different classes, where the continuous variables are measured by the concept of the small bin. We apply mDS on D_{train_H} and $D_{test_H_sORF}$ to rank features for the normal type and sORF type data respectively. We first roughly determine the bin number for each type of data by selecting the bin number as the one that achieves the best accuracy with the least number of features (Table 4.2). The bin number is chosen as 1000 and 100 for normal type and sORF type data correspondingly. We then determine the feature subset sizes according to the accuracy achieved by each feature subset (Fig.4.4). We trained the CNN model using the first n features ranked by mDS, where the interval of n is set as 50 initially and then set as 10 in the interested region. The final sizes of feature subset are determined as 450 ($Acc = 0.8354$) and 566 ($Acc = 0.9707$) for sORF type and normal type data, respectively.

The determined feature subset for sORF type data has a smaller size than the normal type data, indicating that more features are redundant in sORF type data. Our determined feature subsets achieve higher accuracy on both two test datasets than that of the whole feature set. For example, the subset with the first 450 features improves about 2% on accuracy compared to using all 589 features whose accuracy is 0.8153 on sORF type data.

Furthermore, we observe that the features reported to be effective in the last section are ranked at top positions by mDS method. For example, the most efficient feature in normal type transcripts classification, *i.e.* the maximum ORF length feature which achieves over 0.9 accuracy on itself, is sorted as the first important feature in the normal model. And the two powerful features in sORF type transcripts classification, *i.e.* nucleotide bias and hexamer score feature, are both sorted as the first 10 important features in the sORF model.

Fig. 4.4: Accuracies achieved by different sizes of feature subsets. a. normal type data D_{test_H} , and b. sORF type data $D_{test_H_sORF}$.

Table 4.2: The choices of bin number in mDS feature ranking method and the sizes of the corresponding optimal feature subset on *human* normal type and sORF type test dataset.

Normal Type		sORF Type	
Bin number	Optimal feature subset size	Bin number	Optimal feature subset size
100	582	20	490
500	562	40	510
800	566	60	470
1000	566	80	450
1200	564	100	450
1500	566	120	530
2000	568	150	450
5000	586	200	510

4.3.3 Performance of DeepCPP *human* Model

We first evaluate DeepCPP on 4 previous datasets used in the work of CPPred [62], including a *human* test dataset (normal type) (D_{test_H}), a *human* sORF test dataset ($D_{test_H_sORF}$), a *mouse* test dataset (normal type) (D_{test_M}), and a *mouse* sORF test dataset ($D_{test_M_sORF}$). The two *mouse* test datasets are used as the cross species validation datasets. CPPred is a newly proposed approach in RNA coding potential prediction, it is shown to be superior to some previous tools such as CPC2 [57], CPAT [52], and sORF finder [137]. But CPPred is shown to be less effective than PLEK on its two *human* datasets. The authors explain the reason as the high dependency between the training dataset of PLEK and the CPPred *human* test datasets. However, the comparisons made in the work of CPPred are not fair. Because CPPred is trained and tested on ncRNA, whereas PLEK is trained on lncRNA but tested on ncRNA. To make an objective comparison with PLEK, we filter the 4 test datasets by only retaining the lncRNA samples and removing the mRNAs whose lengths are shorter than or equal to 200 (PLEK only handles sequences with lengths > 200). The comparisons are demonstrated in Table 4.3.

The CPPred performs better than PLEK on *human* test datasets but worse on *mouse* test datasets. DeepCPP shows better performances than both two approaches on all test datasets even though it has fewer dependencies between *human* training and test than that of PLEK. It achieves higher values regarding all evaluation matrices except for the specificity (lncRNA prediction accuracy). The improvements of DeepCPP are especially obvious on the sORF test datasets, it improves the accuracy by 7.98% and 0.62% compared to CPPred and PLEK on $D_{test_H_sORF}$, and 9.83% and 14.21% on $D_{test_M_sORF}$, respectively. Particularly, the improvements in the accuracy of DeepCPP are mainly contributed by the noticeable improvements in sensitivity (mRNA prediction accuracy). DeepCPP achieves 20.81% and 5.32% higher values of sensitivity compared to CPPred and PLEK on $D_{test_H_sORF}$, and 26.22% and 28.35% higher on $D_{test_M_sORF}$, respectively. Although the specificity achieved by the DeepCPP is not the best one on two sORF test datasets, *i.e.* 0.9311 and 0.9280 for *human* and *mouse* separately, its significant improvements on sensitivity and its smaller gap between the accuracies of different classes result in higher performances regarding other evaluation criteria.

We use the McNemar test to compare the DeepCPP and other approaches from the perspective of statistics. In the comparison with CPPred, we use the four original test datasets because we use the same training data as that of the CPPred; In the comparison with PLEK, we use the four filtered test datasets to fill the requirements of PLEK. We calculate the values of χ^2 and p-values (denoted as α) for each statistic. The results are shown in Fig.4.5. We make the null hypothesis that the two models perform same statistically. If $\alpha < 0.05$, our null hypothesis will be rejected and suggest that DeepCPP is superior to the other model. Therefore, the p-values which approximately equal to 0 from the figure indicate that our method is better than CPPred, and better than PLEK when evaluated on *mouse* data statistically.

We next evaluate DeepCPP on our newly constucted test datasets, including a *human* test dataset (normal type) $D_{test_H_new}$ and its sORF test dataset $D_{test_H_newsORF}$, where the samples all newly released and the non-coding RNAs are all lncRNA. We also evaluate 7 other newly proposed approaches which are reported to be superior to early programs like CPC [50], CPAT [52], PLEK [54], FEELnc [59], *etc*, on these two test datasets. These approaches consist of Hugo’s SVM method [60], lncRNAnet [67], mRNN [65], LncADeep [66], RNAsamba [64], LncFinder [61], and CPPred [62]. All of these tools use machine learning models, either Supporting Vector Mechine (SVM) model or deep learning model. And they employed diverse features. We briefly summarize the data source, model, and features used in these methods in Table 4.4 and record their performances as well as the performance of DeepCPP in Table 4.5 and 4.6.

DeepCPP realizes high performances when testing on our newly constructed test datasets and in the comparisons with other recently proposed prediction tools which are trained using data from different sources. When evaluating on $D_{test_H_new}$, DeepCPP achieves a little bit lower (0.26% lower) accuracy than LncADeep, but it realizes better performances than other tools. When evaluating on $D_{test_H_newsORF}$, DeepCPP shows obvious advantages over other tools. The sensitivity achieved by DeepCPP is 0.8448, which is 4.31% higher than RNAsamba whose value is 0.8017 and more than 10.34% higher than other tools whose values are all below 0.7414. Again, the specificity of DeepCPP on the sORF test dataset is a little bit low compared to other tools, *i.e.* 0.9267, but its significant improvements on sensitivity and the smaller gap between the specificity and sensitivity lead to higher performances regarding other criteria.

	CPPred correct	CPPred wrong		CPPred correct	CPPred wrong		CPPred correct	CPPred wrong		CPPred correct	CPPred wrong
DeepCPP correct	15934	367	DeepCPP correct	961	170	DeepCPP correct	48544	951	DeepCPP correct	1274	269
DeepCPP wrong	225	271	DeepCPP wrong	69	82	DeepCPP wrong	636	899	DeepCPP wrong	90	213
$\chi^2 = 33.58$ $\alpha = 6.84e - 9$ D_{test_H} (lncRNA 8240 : mRNA 8557)											
$\chi^2 = 41.84$ $\alpha = 9.91e - 11$ $D_{test_H_sORF}$ (lncRNA 641 : mRNA 641)											
$\chi^2 = 62.13$ $\alpha = 3.22e - 15$ D_{test_M} (lncRNA 19930 : mRNA 31100)											
$\chi^2 = 88.26$ $\alpha = 0$ $D_{test_M_sORF}$ (lncRNA 1000 : mRNA 846)											
	PLEK correct	PLEK wrong		PLEK correct	PLEK wrong		PLEK correct	PLEK wrong		PLEK correct	PLEK wrong
DeepCPP correct	15055	383	DeepCPP correct	1015	113	DeepCPP correct	39429	4090	DeepCPP correct	1154	375
DeepCPP wrong	332	163	DeepCPP wrong	105	45	DeepCPP wrong	462	1070	DeepCPP wrong	115	186
$\chi^2 = 3.50$ $\alpha = 0.06$ Filtered D_{test_H} (lncRNA 7378 : mRNA 8555)											
$\chi^2 = 0.22$ $\alpha = 0.64$ filtered $D_{test_H_sORF}$ (lncRNA 639 : mRNA 639)											
$\chi^2 = 2889.97$ $\alpha = 0$ filtered D_{test_M} (lncRNA 13954 : mRNA 31097)											
$\chi^2 = 136.90$ $\alpha = 0$ filtered $D_{test_M_sORF}$ (lncRNA 987 : mRNA 843)											

Fig. 4.5: McNemar test applied on DeepCPP and other methods on 4 previous test datasets. Each table tabulates the correctly or wrongly predicted samples by each model, the McNemar test statistic outcomes and the datasets with data amounts for each class are demonstrated below the table. The filtered datasets in the second row only contain RNAs whose lengths > 200.

We pay extra attention to the mRNA whose longest ORF is less than 100 codons. As stated above, there are 48 such mRNAs in our new test dataset. DeepCPP identifies 36 out of 48 samples as coding RNA, while other approaches only precisely predict at most 34 such samples (Table 4.7).

4.3.4 Performance of DeepCPP Integrate Models

We construct two integrated models using the data from vertebrate and insect species. We use the same feature subset determined in the *human* model for two integrated models. Two approaches, *i.e.* Hugo's SVM method [60] and CPPred [62], also trained the integrated model using species of *mouse* and *zebrafish* from the taxonomic group of vertebrate, and DeepCPP trained the integrated model using species of *fruit fly* and

Table 4.3: Comparison between DeepCPP and other approaches on test datasets from the work of CPPred. The number of samples for the filtered test datasets are shown.

Datasets (#lncRNA:#mRNA)	Methods	Sp	Sn	Acc	F-score	Mcc	Hm
D_{test_H} (7378:8555)	DeepCPP	0.9847	0.9553	0.9689	0.971	0.938	0.970
	CPPred	0.9672	0.9537	0.9600	0.962	0.920	0.960
	PLEK	0.9789	0.9544	0.9657	0.968	0.932	0.966
$D_{test_H_sORF}$ (639:639)	DeepCPP	0.9311	0.8341	0.8826	0.877	0.769	0.880
	CPPred	0.9797	0.6260	0.8028	0.760	0.647	0.764
	PLEK	0.9719	0.7809	0.8764	0.864	0.757	0.866
D_{test_M} (13954:31097)	DeepCPP	0.9733	0.9627	0.9660	0.975	0.922	0.968
	CPPred	0.9679	0.9552	0.9591	0.970	0.907	0.962
	PLEK	0.9061	0.8762	0.8855	0.914	0.751	0.891
$D_{test_M_sORF}$ (987:843)	DeepCPP	0.9280	0.7272	0.8355	0.803	0.675	0.815
	CPPred	0.9696	0.4650	0.7372	0.620	0.514	0.619
	PLEK	0.9068	0.4437	0.6934	0.571	0.401	0.596

Table 4.4: Summary of seven newly published lncRNA and mRNA distinction programs.

Program	Training data source	Feature	Method
Hugo's SVM	Ensembl	K-mer k=2-4, length and relative length for the first and the longest ORF	SVM
mRNN	GENCODE release 25	One-hot vector	GRU network
lncRNAnet	GENCODE release 25	the longest ORF length and coverage, ORF indicator, one-hot vector	RNN
LncADeep	RefSeq release 75, GENCODE release 24	the longest ORF length and coverage, entropy density profile of ORF, mean hexamer score, UTR coverage, Fickett score, GC content, HMMER index	Deep belief network
LncFinder	GENCODE, Ensembl	the longest ORF length and coverage, log-dist hexamer, Secondary structure, physicochemical feature	SVM
RNAseer	GENCODE, Refseq	IGLOO-learned feature, k-mer k=1-4, the longest ORF length	Deep learning
CPPred	NCBI RefSeq, Ensembl	the longest ORF length and coverage, ORF integrity, Hexamer score, Fickett score, isoelectric point, gravity, instability, global transcript sequence descriptors	SVM

Table 4.5: Comparison between DeepCPP *human* model and other approaches on newly constructed *human* test dataset.

	Sp	Sn	ACC	F-score	MCC	Hm
DeepCPP	0.9844	0.9647	0.9820	0.930	0.920	0.974
Hugo's SVM	0.9954	0.0840	0.8828	0.150	0.219	0.155
mRNN	0.9512	0.9665	0.9531	0.836	0.819	0.959
lncRNAnet	0.9783	0.9579	0.9758	0.907	0.895	0.968
LncADEEP	0.9853	0.9792	0.9846	0.940	0.932	0.982
LncFinder	0.9603	0.9621	0.9605	0.858	0.842	0.961
RNAseam	0.9406	0.9816	0.9457	0.817	0.802	0.961
CPPred	0.9602	0.9546	0.9595	0.853	0.836	0.957

Table 4.6: Comparison between DeepCPP *human* model and other approaches on newly constructed *human* sORF type test dataset.

	Sp	Sn	ACC	F-score	MCC	Hm
DeepCPP	0.9267	0.8448	0.8858	0.881	0.774	0.884
Hugo's SVM	0.9957	0.0603	0.5280	0.113	0.158	0.114
mRNN	0.9483	0.6983	0.8233	0.798	0.668	0.804
lncRNAnet	0.9784	0.5862	0.7823	0.729	0.614	0.733
LncADEEP	0.9784	0.7414	0.8599	0.841	0.741	0.844
LncFinder	0.9828	0.5086	0.7457	0.667	0.558	0.670
RNAseam	0.9483	0.8017	0.8750	0.865	0.758	0.869
CPPred	0.9914	0.4828	0.7371	0.647	0.551	0.649

Table 4.7: The number of correctly predicted mRNA samples with ORF lengths less than 100 codons by different approaches.

Method	Correctly predicted samples in 48 mRNA samples
LncFinder	1
Hugo’s SVM method	2
CPPred	16
lncRNAnet	23
LncADeep	26
mRNN	29
RNAseam	34
DeepCPP	36

nematode from the taxonomic group of insect. Therefore, these tools are also evaluated and compared. The performances for different approaches are demonstrated in Fig.4.6.

Hugo’s SVM performs poorly on vertebrate datasets which predict most transcripts to the non-coding type. DeepCPP achieves better performances than CPPred, especially on sORF type data for both vertebrate and insect datasets. DeepCPP improves 37.24% and 17.09% regarding the values of sensitivity and accuracy compared to CPPred, and improves more than 25% on the values of F-score, MCC, and Hm on $D_{test.V.sORF}$. And it improves 0.36% and 3.14% on accuracy than CPPred on $D_{test.I}$ and $D_{test.I.sORF}$ separately. However, when examining the prediction results for each species in vertebrate datasets, we find that fewer ncRNA test samples for *zebrafish* are correctly predicted by both methods. DeepCPP correctly predicts 75.92% ncRNA samples in *zebrafish* and CPPred correctly predicts 58.25%. We use CD-HIT [129] to examine the dependency between *zebrafish* ncRNA training and test samples. Only 11 out of 951 samples in the test dataset are found to have over 0.8 dependencies compared to the training data. The small dependencies between training and test dataset in *zebrafish* ncRNA could explain the relatively lower performances on these data. Our finding also indicates that the intrinsic characteristics of the newly released ncRNA samples show the obvious differences from the samples in early releases in Ensembl *zebrafish* databases.

We also test our DeepCPP vertebrate model on the previous *mouse* datasets ($D_{test.M}$ and $D_{test.M.sORF}$), and test DeepCPP insect model on samples from seen (*fruit fly* and

mosquito) and unseen (*honeybee*) species separately, where the seen and unseen species are distinguished according to if the data from that species is adopted to train the insect model. The comparisons with CPPred are shown in Fig.4.6.e. DeepCPP realizes values of 0.9951, 0.9910, and 0.9926 regarding the specificity, sensitivity, and accuracy on D_{test_M} , and 0.9730, 0.9374, and 0.9567 on $D_{test_M_sORF}$, respectively. It improves substantially compared to CPPred on $D_{test_M_sORF}$ with 46.93% and 21.67% higher values of sensitivity and accuracy separately. DeepCPP realizes 2.4% (normal) and 10.31% (sORF) higher values of accuracy than CPPred on data from species of *fruit fly* and *mosquito*, but its performances on data from species of *honeybee* are worse. One possible reason of the less powerful performance of DeepCPP on *honeybee* mRNA samples is the protein dissimilarity between *honeybee* and the other two insect species used as training. Study shows that the number of proteins and their functions for species of *honeybee* are quite dissimilar to that of *mosquito* and *fruit fly* [138]. Therefore, the mRNAs for species of *honeybee* could also be dissimilar to that of *mosquito* and *fruit fly*, which are indicated by the features derived from the sequences.

4.4 Chapter Summary

In this chapter, we develop DeepCPP to distinguish lncRNA from mRNA by assessing the coding potential of transcripts. The main novelties of DeepCPP including that it considers a wide range of features where some of them were not used in this prediction task before and it contains models targeting to sORF type data specifically. We examine different types of features and demonstrate that our newly proposed nucleotide bias feature contributes to this prediction problem. We pay extra attention to the RNAs with sORFs which used to be a bottleneck by constructing a specific model for this type of data using a different feature subset from that of the normal type data. Our results show that DeepCPP works effectively on different species and it outperforms other recently published approaches. Specifically, the notable improvement of prediction accuracy on mRNA with sORF illustrates the significance of DeepCPP in this prediction problem. After successfully annotating the lncRNA, in the next chapter, we will focus on the lncRNA function exploration with the assistance of machine learning.

Fig. 4.6: Evaluation of DeepCPP integrated models. a-d. Comparisons among DeepCPP, CPPred, and Hugo's SVM method on vertebrate and insect datasets a. D_{test_V} , b. $D_{test_V_sORF}$, c. D_{test_I} , and d. $D_{test_I_sORF}$. e. Comparisons between DeepCPP and CPPred on specific test datasets. From left to right: D_{test_M} , $D_{test_M_sORF}$, normal type data from species of *fruit fly* and *mosquito*, normal type data from species of *honeybee*, sORF type data from species of *fruit fly* and *mosquito*, and sORF type data from species of *honeybee*. CPPred - Integrated model and hugo's SVM - GRCm38.GRCz10 model are adopted for the comparisons.

Chapter 5

Predicting the Types of Biomolecules Interacting with lncRNAs: An Ensemble Deep Learning Approach¹

5.1 Introduction

Long non-coding RNA (lncRNA) is an enormous group of RNAs whose nucleotide length is longer than two hundred. lncRNAs are proved to execute noteworthy functions in complicated pathological and physiological activities [140], hence receiving widespread interests from researchers. A vast amount of lncRNAs have been identified, but only a few of them are adequately studied [10,11]. Unveiling the functional significances of lncRNAs remain to be a challenging task both biologically and computationally considering their complex molecular mechanisms.

Previous researches have shown that lncRNA takes part in a great variety of regulatory functions, including epigenetic modification, gene transcription, chromatin remodeling,

¹The work in this chapter has been published in [139].

translation, and protein function [10–13]. The lncRNAs execute their functions mainly through interacting with other biomolecules.

lncRNA interacts with multiple categories of RNAs, both coding and non-coding ones. lncRNA regulates the expression of coding RNA (mRNA) either in a direct way or in an indirect way. For one thing, lncRNA regulates the editing, stability, splicing, and subcellular distribution of mRNA by targeting it directly [141]. The lncRNA PR11-714G18.1 represses the vascular cell migration of both *human* endothelial cells and vascular smooth muscle cells by binding with the protein-coding gene LRP2BP in a direct way [142]. For another, lncRNA indirectly regulates mRNA through competing endogenous RNA (ceRNA) networks via the shared miRNA responsive elements (MRE) [143,144]. The lncRNA linc-MD1 crosstalks with MAML1 and MEF2C to regulate their expression through shared miR-133 and miR-135, hence governing the muscle differentiation [145]. Another indirect way for lncRNA to regulate the mRNA expression is to compete for the mRNA binding sites with an miRNA. One proof of this mechanism is that lncRNA BACE1-antisense veils the mRNA binding locus for miR-485-5p to avoid the miRNA-induced BACE1 inhibition [146].

lncRNA takes part in interacting with diverse non-coding RNAs (ncRNAs). First, lncRNA serves as precursors of small ncRNAs like piRNA [147], snoRNA [148], and miRNA [149]. Second, the expression and stability of lncRNA can be regulated by miRNA [150], *e.g.*, miRNA MiR-34a downregulates the expression of lncRNA HOTAIR in PCa cell lines by directly targeting it [151]. Third, lncRNA associates with snoRNA. sno-lncRNA denotes a type of lncRNA whose ends are corresponding to the locations of intronic snoRNAs [152]. Evidence has shown that sno-lncRNAs can modify splicing patterns in cells and they are firmly engaged in the splicing regulators from the Fox family [152]. Fourth, lncRNA interacts with other lncRNAs to be involved in allelic and locus-specific control to regulate the epigenetics. Studies have suggested that the lncRNA Jpx establishes a dynamic balance between activators and repressors by activating lncRNA Xist, hence guaranteeing the X-chromosome inactivation (XCI) [153]. Finally, some lncRNA-snoRNA interactions like the MALAT1-U1 are unveiled by the experiments of Psoralen Analysis of RNA Interactions and Structures in both *human* and *mouse* cells [154]. Some popular lncRNA-RNA interactions are briefly demonstrated in Fig.5.1.

lncRNA implements function through the association with proteins by serving as signals, guides, scaffolds, and decoys [155]. First, lncRNA can be induced to respond to stimuli and serves as an indicator that demarcates time, space, and gene expression [156]. The lncRNA CCND1 responds to DNA damage signals to recruit protein TLS into the promoter of CCND1, thereby modulating the target activities [157]. Additionally, lncRNA guides the recruitment of proteins onto chromatin sites, thus regulating gene expression. For example, lncRNA DIGIT guides the recruitment of protein BRD3 to the locus of the chromatin H3K18ac, such interaction in the specific region regulates the differentiation of endoderm with the assistance of the transcription factors [158]. Moreover, lncRNA acts as a protein scaffold. The lncRNA TUG1 associates with proprotein convertase 2 (Pc2) as a scaffold to activate the milieu of interchromatin granules and relocate growth control transcription units, thereby regulating the gene expression [159]. Furthermore, lncRNA serves as decoy to proteins that are away from the binding sites. For example, the lncRNA PANDA inhibits the apoptotic gene expression via interacting with the transcription factor NF- κ B, thus avoiding being associated with the promoters of the target gene [160].

Some new functions of associations between lncRNA and protein have also been unveiled. First of all, lncRNA-protein interaction stabilizes the lncRNA, at the same time, the lncRNA inhibits the binding of protein with the target mRNA by acting as ceRNA. The lncRNA MACC1-AS1 binds with the protein PTBP1, their binding not only stabilizes the MACC1-AS1 but also restricts the binding of PTBP1 to mRNA [161]. Moreover, lncRNA modulates the protein expression either in a direct way or in an indirect way. The lncRNA NEAT1 binds the protein PINK1 in a direct way to enhance its expression, which further regresses the degradation of protein [162]. And the lncRNA RPSAP52 serves as a sponge for an HMGA2-targeted miRNA when it is overexpressed, thereby increasing the protein levels of HMGA2 in an indirect way [163].

lncRNA is also involved in the association with DNA. There are 3 main mechanisms for lncRNA to interact with DNA, including associating itself with DNA-binding proteins, forming R-loops, and forming triplexes. First of all, during the lncRNA-mediated recruitment, the lncRNA targets the DNA sites to either restrict or enhance the attachment of DNA-binding proteins [12, 164]. The lncRNA AFAP1-AS1 interacts with protein

AUF1, which activates the gene translation by enhancing the AUF1-ERBB2 gene association level [165]. Moreover, the formation of the R-loop contributes to gene expression. A study shows that the interaction between lncRNA APOLO and DNA in the form of R-loop modulates the chromatin loops both *in cis* and *in trans*, thus promoting the gene transcription [166]. This study also reveals that the singularities of R-loops in chromatin function and compaction are partly associated with the function of lncRNAs in gene transcriptional regulation and genome organization [166]. Finally, the lncRNA also regulates the gene expression via the formation of DNA:lncRNA triplex. One proof is that the lncRNA MIR100HG forms triplex with DNA to repress the expression of the protein p27, which results in the increment of cell proliferation in triple-negative breast cancer [167]. The 3 main pathways for the lncRNA and DNA interactions discussed above are demonstrated in Fig.5.2.

Revealing the relationship between lncRNA and other biomolecules can shed light on the understanding of the diseases as some of these interactions are connected with diseases [168]. A great variety of *in silico* tools have been presented to unveil the functions of lncRNAs according to their interactions with other biomolecules. These approaches focus on predicting the interaction for a given pair of lncRNA and a specific biomolecule in specified interactions. For instance, GONMF predicts the co-expression in lncRNA-mRNA [169], CIRNN identifies the association between lncRNA and miRNA [73], HLPI-Ensemble detects the specific interaction between lncRNA and protein [74], and a deep learning program is developed to reveal the binding between lncRNA and DNA [78]. Some recent studies provide more thoroughly reviews about the computational approaches in such predictions that can be referred to [10, 12, 170].

However, one concern is that running all above mentioned computational tools would be inappropriate to determine the functions for a newly discovered lncRNA. For one thing, it is time-intensive to run all tools by evaluating the interaction possibilities between the new lncRNA and all molecules from that type of interaction. For another, if a lncRNA would not be able to interact with a type of biomolecule but we run that type of interaction, there would be more chance to make the mistake. Therefore, if we can predict the possible interacting biomolecule types for a lncRNA to narrow down the range of prediction, the efficiency to identify the lncRNA functions could be improved

and the probabilities to make mistakes can be reduced to some degree. Moreover, if we can find any specificities from the interactions between lncRNAs and different types of biomolecules is less explored by previous studies. Hence we ask that, can we identify the types of biomolecules that a lncRNA would interact with? Will there be any specificities exist in lncRNAs when involved in different interactions?

In this work, we expect to identify the types of biomolecules that a lncRNA potentially interacts with, mainly in the interaction with DNA, RNA, and protein. We download data from two databases that collect the interaction information of lncRNAs from the peer-viewed publications. Based on the collected information and taking the data amount into consideration, we divide the lncRNAs into 4 classes, including lncRNA interacting with i) only RNA, ii) only protein, iii) RNA and protein, and iv) DNA-related, where the DNA-related type of lncRNA consists of 4 types of lncRNAs according to their interactions with i) only DNA, ii) DNA and RNA, iii) DNA and protein, and iv) DNA, RNA, and protein. We briefly summarize the interaction mechanisms in our collected data. And we develop lncIBTP (lncRNA Interaction Biomolecule Type Prediction) to identify the possible interaction biomolecule types for a lncRNA. lncIBTP achieves an average value of 0.704, 0.610, 0.565, and 0.719 on the removed redundancy datasets regarding the accuracy, F-score, precision, and recall for 5-fold cross-validation, respectively. When testing *mouse* samples via the cross-species validation, our model achieves an accuracy of 0.517. lncIBTP could help to improve the efficiency for characterizing the functions for newly identified lncRNA by narrowing down the range of performing prediction tools. Besides, according to our collected information and the prediction outputs from lncIBTP, we hypothesis that the attributes for lncRNAs interacting with DNA may be dissimilar to those interacting with only RNA.

5.2 Materials and Methods

5.2.1 Data Description

NPInter v4.0 [171] and lncRInter [172] collected experimentally verified functional interactions for lncRNA which are reported by the peer-viewed publications. We only

Fig. 5.1: Some main interaction mechanisms in the interactions of lncRNA-RNA. a. lncRNA serves as the hosts to some small ncRNAs. b. lncRNA binds to RNA in a direct way. c. lncRNA involved in the ceRNA network to regulate mRNA through miRNA. d. lncRNA competes with miRNA on mRNA binding sites.

Fig. 5.2: The main interaction mechanisms in interactions of lncRNA-DNA. a. lncRNA binds to DNA through the DNA-binding protein. b. lncRNA forms R-loop with DNA. c. lncRNA forms triplex with DNA.

considered the interaction of lncRNA with RNA, protein, and DNA. Therefore, there are 7 possible interactions for a lncRNA, including lncRNA that interacting with i) only RNA, ii) only protein, iii) only DNA, iv) both RNA and protein, v) both DNA and protein, vi) both DNA and RNA, and vii) all of DNA, RNA, and protein. We used a Venn diagram to illustrate the amount of lncRNA genes in each group divided above for two databases (Fig.5.3.a). None of the lncRNA was reported to interact with RNA and DNA simultaneously without interacting with protein. The amounts of lncRNAs from the type of iii, v, vi (no such samples), and vii are much smaller than the other types, therefore, we combined the lncRNAs from these 4 types and named the new type as DNA-related. We integrated the information from both two databases to construct our dataset. The variants that are annotated by the GENCODE release 33 for each lncRNA gene were taken into consideration [173]. lncRNA MIR100HG that was reported to associate with DNA by forming triplex was also considered [167]. Our dataset consisted of 4 classes of lncRNA according to their reported interaction, including lncRNA that interacting with i) only RNA, ii) only protein, iii) RNA and protein, and iv) DNA-related (only DNA, DNA and protein, and DNA, RNA, and protein). We constructed a removed redundancy dataset from the above dataset by using CD-HIT with a threshold of 0.8 [174]. The amounts of lncRNA samples before and after removing redundancies in each class were illustrated in Fig.5.3.b.

A *mouse* dataset was further constructed for the purpose of cross-species validation. The lncRNA interaction information was collected from the NPInter v4.0 database [171] and variants of the lncRNA were contrasted with GENCODE release m24 [173]. The classification of lncRNA samples followed the same criteria as that of the *human*. The number of lncRNA samples in each classes of *mouse* dataset was illustrated in Fig.5.3.c.

5.2.2 Feature Extraction

k-mer features represent the frequencies of k neighboring components in a sequence [175]. The g-gap and g-bigap features count the frequencies of 2-mer and 4-mer with g gaps in the middle of their nucleotides, *e.g.* $A(g)A$ and $AA(g)AA$. Their calculations can be found from Section 4.2.

Fig. 5.3: Construction of Dataset. a. The Venn diagram of lncRNA in two databases according to their reported interacted biomolecules. The numbers of lncRNA genes in each type of interaction for each database are shown. lncRNAs interacting with multiple types of biomolecules are shown as the overlap. Left: NPInter v4.0 database. 5 types of lncRNAs are found in this database based on their interactions with i) only RNA, ii) only protein, iii) RNA and protein, iv) DNA and protein, and v) DNA, RNA, and protein. Right: lncRInter database. Besides the same 5 types of lncRNAs found in NPInter database, lncRInter database has one more type of lncRNA: interacting with only DNA. b. The constructed dataset which integrating the information from lncRInter and NPInter v4.0 databases. The lncRNAs are finally divided into four classes according to their reported interactions with i) only RNA, ii) only protein, iii) RNA and protein, and iv) interacting with DNA-related (only DNA, DNA and protein, and DNA, RNA, and protein). The numbers of lncRNA samples in each class are shown as annotated by GENCODE release 33 / after removing redundancy with a cut-off of 0.8. c. The *mouse* dataset. The lncRNAs are collected from NPInter 4.0 and the numbers of lncRNA samples in each class are shown after contrasting with GENCODE m24 annotation.

Fig. 5.4: The performances of different feature sets evaluated by the model trained with one CNN layer and fifty epochs.

To select the proper values of k and g to be used in this study, we tested different feature sets and their combinations on our data. We randomly split our dataset into training and validation with a ratio of 8:2. We adopted a 1-layer Convolutional Neural Network (CNN) model, and we trained the model with a balanced strategy to avoid the prediction bias considering the relatively large differences in the data amounts for each class. We randomly selected 500 samples from each class in the training data either by over-sampling with the bagging strategy or by down-sampling. The prediction results with different feature sets trained with 50 epochs were demonstrated in Fig.5.4. We determined our final feature set as the one that realizes the highest accuracy from all candidate sets. Therefore, our final feature set consisted of 576 features with 3-mer, 2-bigap, and 3-bigap representation methods whose number of features are 64, 256, and 256, respectively.

5.2.3 Model Construction

There is a large gap between the amount of data in each class of our dataset, the amounts of lncRNA samples in the class of interacting with only protein, and RNA and Protein are around twelve times as that of the other two classes. Therefore, we adopted a balanced training strategy to avoid the potential bias in prediction. We randomly selected 2000 samples from the class of protein and RNA and Protein which have more than 2500 samples, and we over-sampled 2000 samples using the bagging strategy from the class

Fig. 5.5: The construction of IncIBTP. Each model includes 4 CNN layers, 1 flatten layer, 1 dropout layer with dropout rate of 20%, 1 dense layer with activation function of ReLU, another 1 dropout layer with dropout rate of 20%, and another 1 dense layer with activation function of ReLU. Class 0 to 3 are corresponding to lncRNA samples that interacting with only RNA, only protein, RNA and protein, and DNA-related, respectively.

of RNA and DNA-related. To determine the model structure and parameters, we tested one candidate value for one parameter each time on the same randomly split validation dataset (Appendix C.1). The structure and parameters that realized the highest accuracy were chosen for the construction of the final model. Due to the different initializations when training a CNN model, we ensemble 5 models into one and make the decision according to the sum of the output probability scores of the 5 models (Appendix C.2). We re-generated the samples to train each model by over-sampling or down-sampling from the same original training data. We named our model as IncIBTP (lncRNA Interaction Biomolecule Type Prediction) and its structure and detailed parameters were demonstrated in Fig.5.5.

5.2.4 Evaluation Criteria

We assessed the model performance using the macro-average for this four-class classification problem. The criteria for assessment consist of the overall Acc, the macro-average F-score, the macro-average precision, and the macro-average recall, as defined in Chapter 2.4, where the macro-average takes the average from each class which is evaluated independently under each criterion.

5.3 Results and Discussion

5.3.1 Interaction Mechanisms between lncRNA and Biomolecules

To have a better understanding of the lncRNA interaction mechanism, we visualize the subtypes of RNA that interacting with lncRNAs and the mechanisms in the lncRNA-biomolecule based on the information collected from the NPInter v4.0 database [171]. Most lncRNA associates with miRNA, some lncRNA interacts with mRNA, only quite a few of lncRNA ($< 1.5\%$) has been reported to interact with other non-coding RNAs (Fig.5.6.a). lncRNA has been suggested to interact with RNA via multiple kinds of mechanisms, whereas the lncRNA mainly interacts with DNA via binding and/or regulatory (Fig.5.6.b). Some lncRNA has only been reported to interact with a single biomolecule but others have been shown to associate with a wide range of biomolecules (Fig.5.6.c). For example, the lncRNA HELLPAR, AC007319.1, XIST, and NEAT1 have been found to interact with multiple molecules of RNA, proteins, RNA and protein, and RNA and DNA, respectively. Fig.5.6.c also indicates that proteins like WDR33 can interact with diverse lncRNAs.

5.3.2 Performances of lncIBTP

We evaluate the performances of lncIBTP as well as other baseline approaches on our removed redundancy dataset via the 5-fold cross-validation, where each fold from the random split would be used as test data in each validation. The baseline models including Supporting Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors

Fig. 5.6: A summary of the interaction mechanisms between lncRNAs and other biomolecules based on *human* data from NPInter v4.0 database. a. The ratio of RNA types to interact with lncRNAs, where the miRNA and mRNA take up the majority parts. b. The number of the cases for different interaction modes in different biomolecules associated with lncRNAs, representing in the form of log2. c. Some lncRNAs are shown to associate with multiple molecules whereas some only are reported to associate with a single molecule. Color-molecule: purple-lncRNA, red-RNA, green-protein, blue-DNA. The black straight line denotes the interaction.

(KNN), whose parameters are determined as the ones that achieving the optimal accuracy, implemented by the Python function of *sklearn.model_selection.GridSearchCV(estimator,cv=5)* (Appendix C.3). The results are shown as the box and whisker plot in Fig.5.7.a, where the model skew is reflected.

Our method shows a general good prediction performance. The values of the accuracy range from 0.688 to 0.724, and the values of the macro-average F-score are in the range of 0.569 to 0.677. The generally small deviation regarding the accuracy reflects the model’s robustness. We notice that even the single CNN model could work more efficiently than other baseline methods. The values of the average accuracy and precision are 0.674 and 0.647 for single CNN but are less than 0.646 and 0.609 for other baseline models.

To assess the performance of IncIBTP on each class, we visualize the confusion matrix (Fig.5.7.b), the ROC Curves, the Precision-Recall Curves (PRC) (Fig.5.8), as well as the corresponding values of Area Under the ROC curve (AUROC) and Area Under the PRC curve (AUPRC) (Table 5.1) from each validation. We use class 0-3 to refer to the classes of lncRNAs that interacting with RNA, protein, RNA and protein, and DNA-related. Most samples are predicted to class 1 or class 2. The accuracies for two major classes (class 1 and class 2) which have more data amount are higher than that of the two minor classes (class 0 and class 3). Interestingly, although the prediction accuracy for class 3 is lower than that of class 1 and class 2, its AUROC values are not always worse than the two major classes. The AUROC value for class 3 is larger than that of the two major classes in 4 out of 5 folds, and it is approximate to the 2 major classes in the remaining one fold. The average AUROC values for class 0-3 are 0.723, 0.805, 0.799, and 0.834, respectively, where class 3 realizes the largest value.

The same tendencies are also supported by the ROC and PRC curves. The good performance of class 3 regarding the ROC and PRC curves can be explained from its small data amount and much less mispredicted negative (the other three classes) samples. For class 3, its $TP + FN$ is far less than its $FP + TN$, according to the calculation of True Positive Rate (TPR) ($TPR = \frac{TP}{TP+FN}$) and False Positive Rate (FPR) ($FPR = \frac{FP}{FP+TN}$), it can be deduced that the influence of the increment of TP on TPR can be more notable than that of FP on FPR. Therefore, in the ROC curve, the TPR value of class 3 is larger than other classes when the FPR values are the same and small. The above analysis

Fig. 5.7: The performances of lncIBTP. a. The comparison between lncIBTP and other baseline methods via the 5-fold cross-validation represented by the boxplots. b. The confusion matrix using each fold as test data in 5-fold cross-validation. x-axis and y-axis denote the predicted label and the true label separately. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively.

also indicates that the threshold to separate the positive and negative samples for class 3 can be small. Similarly, the precision value of class 3 remains at 1 before its recall value exceeding a certain value in the PRC curves, that is $TP = 1$ and $FP = 0$ before its FN reaching a threshold according to equation ?? and ??. This phenomenon indicates that no negative samples would be wrongly predicted when the FN is less than a certain value.

We also make other observations from the confusion matrix (Fig.5.7.b). For example, the samples from other classes are less likely to be predicted to class 3. Particularly, we observe that the samples in class 0 and class 3 are hard to be predicted to each other. Recalling the Fig.5.3.a which shows the lncRNA interaction information from NPInter and lncRInter databases that no lncRNAs are reported to interact with DNA and RNA simultaneously without interacting with protein, we speculate that the attributes of the lncRNA that interacting with DNA would be dissimilar to those only interacting with RNA based on the limited data.

Fig. 5.8: The illustration of the ROC (left) and PRC (right) curves for each class in the 5-fold cross-validation.

Table 5.1: The AUROC (Area Under the ROC curves) and AUPRC (Area Under the PRC curves) values for each class in the 5 folds from cross-validation.

class	ROC				PRC			
	RNA	protein	RNA & protein	DNA related	RNA	protein	RNA & protein	DNA related
Fold 1	0.7376	0.8160	0.8102	0.8576	0.3133	0.7908	0.7896	0.7200
Fold 2	0.7207	0.7855	0.7790	0.8182	0.2912	0.7708	0.7361	0.6640
Fold 3	0.7000	0.7978	0.7941	0.8542	0.2353	0.7941	0.7618	0.6885
Fold 4	0.7872	0.8166	0.8036	0.8443	0.4790	0.8133	0.7722	0.7114
Fold 5	0.6715	0.8067	0.8093	0.7967	0.2924	0.7872	0.7906	0.6410
Average	0.7234	0.8045	0.7992	0.8342	0.3222	0.7912	0.7700	0.6850

5.3.3 IncIBTP as a Pre-step for Specific lncRNA-molecule Prediction

RNAInter is an RNA interactome database [176]. It provides both the experimental evidence and the prediction score between a lncRNA and a specific molecule in any one type of associations between lncRNA and RNA, protein, and DNA. Therefore, the types of biomolecules that a lncRNA interacts with could be deduced from the corresponding confidential scores output by the RNAInter. We search the confidential scores predicted by RNAInter in all three types of interactions for each lncRNA from each fold in our 5-fold cross-validation datasets. We regard a lncRNA sample to be involved in the interaction with a type of biomolecule if it is found to interact with any molecules from that type of biomolecule with a score larger than 0.5. The confusion matrices for the prediction of RNAInter on each fold are shown in Fig.5.9. The RNAInter performs worse than that of IncIBTP by achieving an average accuracy of 0.535. But when we use IncIBTP as a pre-step for RNAInter, that is we predict the types of biomolecules to interact with a lncRNA firstly, RNAInter achieves an average accuracy of 0.9232 on the correctly predicted samples from IncIBTP.

Fig. 5.9: Confusion matrix of RNAInter prediction. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively.

Fig. 5.10: The reported and the predicted types of biomolecules to be interact with SOCAR, RAIN, LINC02407, and LINC01980.

5.3.4 A Case Study

We evaluate the lncIBTP using several newly identified lncRNAs from 2019 onwards, including RAIN, SOCAR, LINC01980, and LINC02407 [176–180]. These lncRNAs were validated by the experiments to interact with RNA and/or protein but they are not in the existing databases. We obtain their variants from the NCBI nucleotide database which are used as the input of our model. The experimentally verified types of biomolecules and the predicted types of biomolecules by lncIBTP for the lncRNAs collected above are shown in Fig.5.10. The results predicted by lncIBTP are nearly in agreement with the verified interactions. But the prediction results of lncIBTP suggest that the RAIN also can be associated with RNA and LINC02407 also can be associated with protein, which provides some insights for the feature exploration.

5.3.5 Cross-species Validation

We also evaluate the lncIBTP via a cross-species validation, where the model is trained with all *human* samples but evaluated on a *mouse* dataset. The accuracy is 0.517, which is a little bit low compared to the validation results on *human* data. One potential explanation of this phenomenon is that the features from *mouse* data may be different from the *human* data. We show the feature heatmaps for both *human* and *mouse* species in Fig.5.12) for each class in our dataset. We observe some noticeable differences between the two species. For example, the position at row 1 column 7 (sequencing from 0) in the heatmap which denoting the average frequencies of ‘CGC’. In the *human* data, the values at this position are 0.0045, 0.0042, 0.0051, and 0.0045 for class 0-3, respectively, but in the *mouse* data, the values are 0.0057, 0.0051, 0.0053, and 0.0082, respectively. The largest value of the average frequencies of ‘CGC’ in *human* data is in class 2 but in *mouse* data is in class 3. Moreover, ‘CGC’ appears more frequently in *mouse* data.

Although the cross-species validation result is not very desirable, we observe similar tendencies like that of the *human* data from the *mouse* confusion matrix (Fig.5.11). For instance, a large number of samples are predicted to two major classes (class 1 and class 2), the prediction accuracy for class 0 is the lowest one, and no sample from class 3 is mispredicted to class 0.

One possible solution to increase the prediction accuracy on *mouse* data is to tune the lncIBTP *human* model parameters. However, this practice is currently impracticable due to the limited data amount in *mouse* dataset. For example, only 24 lncRNA variants are collected in the group of DNA-related.

5.4 Chapter Summary

In this chapter, we explore the lncRNA function from a fresh perspective. We focus on the interactions between lncRNA and RNA, protein, and DNA, and develop an ensemble deep learning model named lncIBTP to infer the possible types of interaction biomolecules for a given lncRNA. We show that lncIBTP is an effective and robust tool that delivers general good prediction results on different test datasets. We also make the hypothesis

Fig. 5.11: The confusion matrix for *mouse* data. Class 0 to 3 are corresponding to lncRNAs interacting with RNA, protein, RNA and protein, and DNA-related, respectively.

Fig. 5.12: Feature heatmaps for 4 classes in *human* and *mouse* dataset. Top: *human* data, bottom: *mouse* data. The heatmap from left to right is corresponding to class of RNA, protein, RNA and protein, and DNA-related, respectively. The original features (ordered as 3-mer, 2-bigap, and 3-bigap with nucleotide order of *ACGT*) are reshaped to size of 32×18 .

that the attributes of lncRNA samples that interacting with DNA may be dissimilar to those interacting only with RNA, based on our collected data and the prediction results with limited data. Furthermore, lncIBTP shows better performances than RNAInter on our datasets. And we demonstrate that using the lncIBTP as a pre-step for specific lncRNA-molecule interaction prediction tools like RNAInter can enhance the efficiency to predict the lncRNA functions. We expect that the lncIBTP can shed light on the investigation of lncRNA characteristics and functions. After determining the types of biomolecules that a lncRNA interacts with, in the next chapter, we will continue to explore the lncRNA function prediction by focusing on a specific type of interaction between lncRNA and DNA, *i.e.* DNA:lncRNA triplexes.

Chapter 6

Deep Learning-based DNA:lncRNA Triplexes Prediction¹

6.1 Introduction

The high-throughput sequencing (HTS) techniques have identified a substantial amount of lncRNAs. lncRNAs carry out a great variety of functions in complex physiopathological and biological activities via the interaction with other biomolecules. For example, they modify epigenetic, regulate gene expression, modulate the structure, state, and function of chromatin, *etc* [10–14].

One approach for lncRNAs to execute functions is to form triplex with DNA [182]. The triplex is formed by inserting a third strand (RNA) into a duplex DNA based on either forward or reverse Hoogsteen base-pairing rule between the Watson-Crick base pair and the inserted base [183]. Over the years, pieces of evidence have been found that the lncRNA carries out functions via the formation of triplex with DNA. For example, an lncRNA associated with promoter has been reported to inhibit the transcription of rRNA by interacting with TTF-I [184], FENDRR promotes the occupancy of PRC2 at the triplex formation sites [185], MEG3 alters the gene activity by binding with TGF- β gene [34], PARTICLE forms triplex with MAT2A promoter CpG island to contribute

¹The work in this chapter has been published in [181].

to silence gene [186], KHPS1 anchors the lncRNA by interacting with SPHK1 and it associates effector proteins to the gene promoter [126], HOTAIR regulates adipogenic differentiation in the formation of triplex with PCDH7 and HOXB2 [187], MIR100HG regulates p27 by forming triplex with it [167], and PAPAS inhibits rRNA synthesis by binding the rDNA promoter and recruiting associated CHD4/NuRD [188].

Some HTS techniques are available to map the lncRNA-chromatin interactions throughout the genome by deciphering the binding sites for specific lncRNAs, including Chromatin Isolation by RNA Purification Sequencing (ChIRP-Seq) [31], Capture Hybridization Analysis of RNA Targets Sequencing (CHART-Seq) [32], RNA Antisense Purification Sequencing (RAP-Seq) [33], and Chromatin Oligo affinity Precipitation Sequencing (ChOP-Seq) [34], *etc.* However, these approaches were mainly implemented in crosslinked chromatin, which indicate that the lncRNAs they identified include those associated with DNA binding proteins. Hence these approaches cannot be used as reliable data sources for DNA:lncRNA triplexes. Fortunately, Cetin *et al.* presented an approach to generate the map of DNA:RNA triplex interactions throughout the genome which eliminating the chromatin crosslinking [189]. They unveiled the presences and proved the physiological relevance of the DNA:RNA triplexes *in vivo*.

Several computational approaches have been presented to detect the DNA:RNA triplexes. These approaches mainly depend on extracting the theoretical relationships based on the base pairing rules between DNA and RNA. For example, Triplexator adopted the base-pairing rules of Hoogsteen and reverse-Hoogsteen to systematically calculate the potentiality of the triplex-forming locus on RNA and DNA [79], Triplex-Inspector considered the genomic architecture and gene location to detect sequence-specific ligands and targets [82], LongTarget used a variety of non-canonical rules to identify triplex-forming motifs and binding sites [80], and Triplex Domain Finder (TDF) considered the canonical codes to identify triplexes in lncRNA and its targeting DNAs [81].

However, the computational tools based on mathematical statistics of canonical rules identify a vast amount of lncRNAs with the potentiality to form triplex. But the evidence from *in vivo* or *in vitro* assays suggests the limited number of lncRNAs to form triplex, which indicates that maybe not all of the lncRNAs predicted by the above computational tools can form triplex in practice. Another limitation of these tools is that they do

not take any experimentally verified data into consideration but only the theoretical relationships.

Therefore, in this study, we aim to predict the most possible lncRNAs to form triplex in practice from the triplex-forming lncRNAs characterized by previous canonical rule-based tools, and output the probability score for a DNA site in forming triplex considering the experimentally identified data. To achieve these two goals, we present a tool called TriplexFPP (Triplex-Forming Potential Prediction). TriplexFPP should be the first machine learning approach in DNA:lncRNA triplexes prediction as far as we can tell. When evaluating TriplexFPP via the 5-fold cross-validation, it realizes an average value of 0.9649 and 0.9996 regarding the Area Under the ROC (AUROC) and Area Under the PRC (AUPRC) on the filtered redundancy dataset with a cut-off of 0.8 in triplex lncRNA prediction, and achieves 0.8705 and 0.9671 on the triplex DNA site potential prediction dataset, respectively. Besides, we summarily list the *cis* and/or *trans* binding information of triplex lncRNAs, which may contribute to the investigation of DNA:lncRNA triplexes.

6.2 Materials and Methods

6.2.1 Dataset

6.2.1.1 Triplex lncRNA Dataset

The lncRNA samples in forming triplex were collected from two sources. For one thing, we collected the lncRNA variants whose regions were reported to form triplex from the work of Sentürk *et al.* [189]. Sentürk *et al.* considered both immunopurification with anti-DNA antibody RNA separation and Solid Phase Reversible Immobilization-based paramagnetic bead size selection to specify lncRNA, and they named the peaks detected to form triplex in RNA as TriplexRNA. We obtained the sequences of lncRNA variants by overlapping the TriplexRNA regions with the GENCODE release 33 annotated lncRNA [173]. 476 samples were collected in such a manner and we termed these samples triplexlncRNA. For another, we manually collected lncRNAs from the peer-viewed

publications which were reported to form triplex with DNA from *in vivo* or *in vitro* assays. We named this type of lncRNA as the reported triplex lncRNA, including PARTICL [186], MEG3 [34], HOTAIR [187], FENDRR [190], and MIR100HG [167]. The annotated variants from GENCODE release 33 for the reported triplex lncRNA were downloaded, whose amount is 159.

Among the samples collected above, we removed those with low triplex-forming capacities evaluated by one of the previous computational methods, *i.e.* TDF [81]. We did this because our purpose is to identify the lncRNAs that are more possibly to form triplex in practice from the triplex lncRNAs detected by those canonical rule-based computational methods. We applied TDF with default parameters on the above collected 635 samples to evaluate their potential to form triplexes with the whole gene promoters (exclude chromosome Y and M). 104 samples do not have powerful Triplex-Forming Oligonucleotide (TFO) supported DNA Binding Domains (DBDs). The possible reasons for this phenomenon including i) we collected all variants for reported triplex lncRNA, but maybe not all of them cover the sites that satisfying the canonical rules strongly enough, and ii) the overlapping regions between TriplexRNA and R-Loop cannot be determined as triplex-forming or not for triplexlncRNA [189]. We deleted these samples with low triplex-forming capacities from our positive dataset, and these samples would not be put into the negative dataset neither. In this way, our finally positive dataset contained 531 lncRNA samples.

The samples in negative dataset were selected as the lncRNA variants with comparable triplex-forming capacities evaluated by the TDF. Similarly, we applied TDF with default parameters on all lncRNA variants from GENCODE excluding those in our original positive dataset with the whole gene promoters. The negative samples were selected as those owning at least one DBD with powerful TFO and no less than 123 DNA Binding Sites (DBSs), as samples in our positive dataset have at least 123 DBSs. We also removed the samples from lncRNA MALAT1 which is found to form RNA-RNA triplex [191] and one sample with the letter ‘N’ in its sequence from our negative dataset. Our final negative dataset consisted of 36021 lncRNA samples whose capacities to form triplex are matched with the positive data.

Based on the datasets constructed above, we built 2 more datasets with fewer dependencies. We applied CD-HIT [174] with the threshold (th) of 0.9 and 0.8 separately

Fig. 6.1: A brief illustration of data collection.

to remove the redundant samples in each class. The numbers of positive and negative samples in 2 new datasets are 384 and 28012 (th=0.9), and 286 and 22681 (th=0.8), respectively.

6.2.1.2 Triplex DNA Site Dataset

Our DNA site samples to form triplex were also collected from the work of Sentürk *et al.* [189]. They enriched their DNAs via an unbiased approach and named the peaks detected to form triplex in DNA as TriplexDNA regions. The final positive dataset contained 2542 samples after removing five samples with the letter 'N' in their sequences. 12735 negative samples (five times of the originally positive samples) were randomly generated at the regions of promoters annotated by Ensembl database [44]. Each sample was generated by choosing a chromosome (except chrY and chrM) and a region at random. The process was repeated 5 times where each time the samples were generated with the same length corresponding to the positive dataset. The sequences for positive and negative samples were both derived from the minus strand of DNA.

The data collection process was briefly illustrated in Fig.6.1.

6.2.2 Feature Extraction

We used the approaches of k-mer and kmerscore to extract features from the raw sequences. k-mer features represent the frequencies of k neighboring components in a sequence [175], whose calculation can be found from Section 4.2. Supposing that we have a sequence $S = CCCCCG$, to obtain the vectors of 3-mer, we counted the frequencies of all 3-mer nucleotide composition, where the values for CCC and CCG are $2/3$ and $1/3$ separately, while the values for other compositions in the vector are all zero.

kmerscore measures the overall bias of the corresponding k-mer components in a sequence from the prior knowledge. We obtained the mean values of k-mer vectors for the positive ($M_{pos}(h_i)$) and negative ($M_{neg}(h_i)$) training dataset separately by averaging the k-mer features for the corresponding training samples, where $i = 1, 2, \dots, 4^k$. The kmerscore for a nucleotide sequence $S = s_1 s_2 \dots s_n$ with k-mer sequence $S = h_1 h_2 \dots h_{n-k+1}$ can be obtained as:

$$kmerscore = \frac{1}{n - k + 1} \sum_{i=1}^{n-k+1} \log \frac{M_{neg}(h_i)}{M_{pos}(h_i)} \quad (6.1)$$

6.2.3 Model Construction

We constructed a triplex lncRNA prediction model and a triplex DNA site potential prediction model to predict triplex-forming potential in DNA:lncRNA triplexes. Both two models were implemented using the Convolutional Neural Network (CNN). Our positive and negative datasets are imbalanced in sizes with ratios of 1:68 and 1:5 for lncRNA and DNA site prediction, respectively. To prevent the prediction bias, we down-sampled the negative training data by randomly selecting the equal number of samples as the positive. We also applied the over-sampling technique on the positive training data of triplex lncRNA prediction due to its small data amount. We over-sampled the triplex lncRNA positive data with different weights regarding the different data sources. Because the ratio between triplexlncRNA and reported triplex lncRNA in our positive dataset is about 2.5:1, the model may learn the features favored by the major type rather than the global ones if we treat them equally. Suppose that our lncRNA positive training dataset has n samples, we first over-sampled the whole positive dataset using the bagging strategy

Fig. 6.2: The construction of TriplexFPP. TriplexFPP contains two individual models that corresponding to triplex lncRNA and DNA sites prediction. The detailed structures and parameters are illustrated in the figure.

to the size of $2n$, we then over-sampled n samples from the reported triplex lncRNA sets extra. We called this augmentation process the weighted bagging strategy.

Based on the above data sampling strategies, we determined the number of convolutional layers and other parameters of each model by testing on the corresponding random split datasets (Appendix D.1). We tested one candidate value from one parameter each time, the parameters that achieved the optimal value of sensitivity (accuracy of positive data) were selected. With the optimal parameters, we presented a deep learning tool named TriplexFPP (Triplex-Forming Potential Prediction) to predict the triplex-forming potential. TriplexFPP was composed of two models to identify the triplex lncRNA and the triplex DNA sites separately, where the detailed architectures and parameters for two individual models were shown in Fig.6.2.

6.2.4 Evaluation Criteria

We used a wide range of criteria as defined in Chapter 2.4 to evaluate the model, including Acc, Sn, Sp, Hm of Sn and Sp, and F-score.

Fig. 6.3: Determining the features for TripLexFPP triplex lncRNA (left) and DNA sites (right) prediction model.

6.3 Results and Discussion

6.3.1 Feature and Data Exploration

Each raw input sample is represented by a fixed-length 90-dim vector using the k-mer ($k = 1, 2, 3$, 84 features) and kmerscore ($k = 1 - 6$, 6 features) strategies. The kmerscore feature sets have a powerful prediction efficiency with only 6 features on both two types of predictions, the overall prediction performances can be further improved by adding the k-mer features (Fig.6.3). Because the kmerscore features considering the knowledge from the training dataset, we recalculate the $M_{pos}(h_i)$ and $M_{neg}(h_i)$ in each validation to ensure that no information from test dataset is included.

We plot the nucleotide compositions of lncRNA samples from our positive and negative dataset, where the lncRNA samples from different sources (triplexlncRNA or reported triplex lncRNA) are also shown (Fig.6.4). The triplexlncRNA samples show obvious different preferences on the nucleotide compositions from the reported triplex lncRNAs and the negative samples. For example, the sequences of triplexlncRNA are mainly *CG* rich but the sequences of the reported triplex lncRNAs and the negative lncRNA samples are mainly *AT* rich. The evidence here shows the need to apply the weighted bagging strategy we proposed in the section above.

Fig. 6.4: The nucleotide frequency distribution in positive data from different sources and negative data in lncRNA triplex prediction datasets.

6.3.2 Performance of Triplex lncRNA Prediction Model

We assess the performance of TriplexFPP in identifying the lncRNAs that are more likely to form triplex in practice from the lncRNAs with triplex-forming abilities determined by TDF via the 5-fold cross-validation, and compare it with baseline approaches such as Deep Neural Network (NN), Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting. The results are demonstrated in Fig.6.6.a using the box and whisker plot. The optimal parameters that achieving the best sensitivity values on the random split datasets are selected for the training of the baseline models (Appendix D.1), where the parameters for the NN model are selected manually and the parameters for other models are determined using the Python function of *sklearn.model_selection.GridSearchCV(estimator, scoring='recall', cv=5)*. The training data for baseline models was determined following the same rules as that of TriplexFPP, that is, the positive training samples are tripled using the weighted bagging strategy and the negative samples are down-sampled as the same amounts as the augmented positive data in each validation.

The Gradient Boosting model achieves the highest accuracy and F1-score, however, its sensitivity ranges from 0.8774 to 0.9245. By contrast, our method realizes the value of sensitivity from 0.9340 to 0.9717. Considering that we aim to identify the lncRNAs

to from triplexes in practice, a good performance on sensitivity is more important than that of the specificity. From this perspective, TriplexFPP is superior to other baseline models. Besides, TriplexFPP achieves an average value of 0.9835, 0.9926, 0.9999, and 0.9920 regarding the accuracy, AUROC, AUPRC, and F1-score, which demonstrate its effectiveness in predicting the lncRNA triplex-forming potential.

The 531 lncRNA samples in our positive dataset are from 135 genes, and 57 genes have at least two variants (Fig.6.5). The variants from the same gene may in high agreement with each other regarding the nucleotide compositions, thus contributing to a great prediction outcome. To assess the effectiveness and robustness of TriplexFPP, we implement the 5-fold cross-validation on 2 datasets with fewer redundancies with $th = 0.9$ and $th = 0.8$ separately, and also execute the leave-out validation to exclude the dependencies between training and test.

The prediction results of TriplexFPP on two removed redundancy datasets are compared with that of the original dataset in Fig.6.6.b. The values of the AUROC and AUPRC located in the range of 0.9637 - 0.9880, and 0.9993 - 0.9998 on removed redundancy dataset with $th = 0.9$, and 0.9497 - 0.9809, and 0.9994 - 0.9998 with $th = 0.8$, respectively. TriplexFPP achieves good performances on these two datasets and it only slightly reduces the performances compared to the results on the original dataset.

We implement the leave-out validation by selecting two lncRNAs with the most number of variants from each source in the positive dataset as the test data. PVT1 and LINC00963 with 73 and 52 variants are selected from triplexlncRNA, and MIR100HG and MEG3 with 87 and 46 variants are selected from the reported triplex lncRNA, respectively. In each leave-out validation, we use all the variants from one selected lncRNA as positive test data and use all other positive samples as the training data. We repeat each validation process 5 times with the corresponding negative data from the above original 5-fold cross-validation dataset.

The leave-out validation results are demonstrated in Fig.6.6.c. Our model correctly predicts all variants of PVT1 and LINC00963 as positive and achieves an average AUROC value of 0.9996 and 0.9968 when using PVT1 and LINC00963 as test data. However, the performances on predicting MIR100HG and MEG3 are a little bit low, whose average AUROC values are 0.6594 and 0.3220, respectively. One explanation for the different

Fig. 6.5: The demonstration of the number of variants for each gene, the light color to dark color means a small variant number to a large variant number, outer: triplexlncRNA, inner: reported triplex lncRNA.

performances is that the triplexlncRNA samples are selected as those overlapping with experimentally identified regions, but the reported triplex lncRNA samples are selected as all the variants for that lncRNA, where some of their variants may not form triplex in practice. Interestingly, when we train the model using only six kmerscore features and test on MEG3, the average AUROC value is 0.7610, but the same phenomenon is not observed when testing on MIR100HG.

6.3.3 Performance of Triplex DNA Site Prediction Model

We next evaluate the TriplexFPP triplex DNA site prediction model via the 5-fold cross-validation and compare it with other baseline approaches (Fig.6.7.a). The parameters for baseline models are selected in the same way as triplex DNA site prediction we described in the last section Appendix D.1, Fig.???. Our model achieves an average of 0.8705 and 0.9671 regarding the AUROC and AUPRC separately, which performs better than other baseline methods whose average values of AUROC and AUPRC ranging from 0.8635 to 0.8667, and 0.9642 to 0.9660, respectively. Although the prediction results of the triplex DNA site potential prediction model are lower than that of the triplex lncRNA prediction, our model predicts most triplex-forming DNA sites correctly with high confidence, whose predicted probability scores concentrating at around value of 1 (Fig.6.7.b and Fig.6.8).

Besides, our model correctly predicts a region in HOTAIR sequence (649-708) as positive which has been verified to form DNA:lncRNA triplex [187]. With the limited

Fig. 6.6: Prediction results of TriplexFPP triplex lncRNA model. a. The comparisons between CNN and other baseline methods (SVM, RF, Gradient Boosting, and NN) via the 5-fold cross-validation represented by boxplots. b. The comparisons among original dataset and datasets with removed redundancy ($th = 0.9$ and $th = 0.8$) via the 5-fold cross-validation. c. Radar chart for leave-out validation.

Fig. 6.7: Prediction results of TriplexFPP triplex DNA sites model. a. The comparisons between CNN and other baseline methods (SVM, RF, Gradient Boosting, and NN) via for the 5-fold cross-validation represented by the boxplots. b. The predicted probability score distribution for validation data from the first fold.

data, TriplexFPP shows the ability to distinguish the triplex-forming DNA sites identified by *in vivo* assay from the background sites.

6.3.4 TriplexFPP Model Interpretation

We show the average values for original features, features trained by 1 CNN layer, and features trained by 2 CNN layers from the first fold validation training data for triplex-forming lncRNA and DNA prediction model separately (Fig.6.9 and Fig.6.10). The first six features in the original feature set which are corresponding to the kmerscore features demonstrate noticeable differences between 2 classes in both two prediction models, which suggests that the positive and negative samples have different preferences of nucleotide compositions at the corresponding positions. The convolution average values with the kmerscore features also show obvious differences after training with only 1 CNN layer. The differences between the two classes regarding the k-mer features are not so noticeable until trained by two CNN layers.

6.3.5 Limitation

There are two main limitations of this work regarding the data. For one thing, some samples in our current negative dataset might belong to the positive but they have not

Fig. 6.8: The distribution of the probability scores predicted by TriplexFPP for the other 4 folds of cross-fold validation in triplex DNA sites dataset.

Fig. 6.9: An interpretation of the TriplexFPP triplex lncRNA prediction model from the perspective of input features, where the kmerscore featurers are more distinguishable between positive and negative samples. The first row represents the average values of the original features which are ordered as kmerscore feature with k from 1 to 6, followed by k -mer features with k from 1 to 3 with the nucleotide order *ACGT*, and then the features were reshaped to a 9×10 matrix; the second and the third row represent the average values of the features obtained by 1 CNN layer and 2 CNN layers separately, where the x-axis and y-axis denote the filter and the 1st to the 15th convolution values, respectively. Left: positive training set, right: negative training set.

Fig. 6.10: An interpretation of the TriplexFPP triplex DNA sites prediction model from the perspective of input features, where the kmerscore features are more distinguishable between positive and negative samples. The first row represents the average values of the original features which are ordered as kmerscore feature with k from 1 to 6, followed by k -mer features with k from 1 to 3 with the nucleotide order *ACGT*, and then the features were reshaped to a 9×10 matrix; the second and the third row represent the average values of the features obtained by 1 CNN layer and 2 CNN layers separately, where the x-axis and y-axis denote the filter and the 1st to the 15th convolution values, respectively. Left: positive training set, right: negative training set.

Table 6.1: The binding information for reported triplex lncRNA.

lncRNA	lncRNA chromosome	Bound DNA name and chromosome [ref]	type
HOTAIR	chr12	PCDH7 chr4	<i>in trans</i>
		HOXB2 chr17 [183, 187]	
MEG3	chr14	TGF-b chr3 [34, 183]	<i>in trans</i>
PARTICL	chr2	MAT2A chr2 [183, 186]	<i>in cis</i>
MIR100HG	chr11	P27(CDKN1B) chr12 [167]	<i>in trans</i>
FENDRR	chr16	FOXF1 chr16 [190]	<i>in cis</i>

been identified yet. For another, a small fraction of samples from TriplexlncRNA may belong to the formation of R-Loop rather than the triplex. Therefore, we look forward to more high-quality data to facilitate the improvement of TriplexFPP.

6.3.6 An Exploration on *in cis* / *in trans* Triplex lncRNA

The DNA:lncRNA triplexes could be formed both *in cis* and *in trans*, however, if there are any differences between the mechanisms of the *cis* and *trans* binding of triplex-forming lncRNAs remain unclear [192]. We briefly show the information of *cis* and *trans* for triplex-forming lncRNAs from our triplex lncRNA prediction positive dataset.

For the samples from triplexlncRNA, we use TDF to indicate their interactions with the samples from TriplexDNA (Fig.6.11). 238 samples are found to target with DNA both *in cis* and *in trans*, whose numbers of *in cis* targeting range from 1 to 2450. The remaining 141 samples only are found to target with DNA *in trans*. We also observed that the variants from the same lncRNA gene could have different targeting modes. 26 out of 130 lncRNA genes contain variants either targeting the DNA both *in cis* and *in trans* or only targeting the DNA *in trans* (Fig.6.12 and Fig.6.13).

For the samples from reported triplex lncRNA, we summarized their targeting details in Table 6.1 according to the corresponding peer-viewed publications [34, 167, 183, 186, 187, 191]. lncRNA MEG3, HOTAIR, and MIR100HG are indicated to bind with DNA *in trans*, whereas FENDRR and PARTICLE are indicated to bind with DNA *in cis*.

Fig. 6.11: Statistic of *in cis* & *in trans* lncRNAs in TriplexlncRNA evaluated by TDF. Left: the number of lncRNAs in 2 types of targeting; right: the amounts of *in cis* binding sites for *in cis* & *in trans* targeting lncRNAs.

Fig. 6.12: Statistic of *in cis* & *in trans* targeting lncRNAs in TriplexlncRNA evaluated by TDF. Left: gene numbers in each type of targeting, right: variant numbers of PVT1 and LINC00963 in two types of targeting.

Fig. 6.13: Gene names and number of variants for lncRNAs in two types of targeting.

6.4 Chapter Summary

In this chapter, we propose a deep learning tool named TriplexFPP in DNA:lncRNA triplexes prediction. TriplexFPP predicts the lncRNAs that are more possibly to form triplex in practice from the lncRNAs that are theoretically predicted to form triplex by the canonical rule-based computational methods, and it also predicts the possibility for a DNA site to form triplex. It narrows down the range of triplex lncRNAs detected by previous approaches and it takes the data from either *in vivo* or *in vitro* assays into consideration. The good prediction results of TriplexFPP demonstrate its effectiveness and robustness in DNA:lncRNA triplexes prediction. We expect the TriplexFPP can contribute to the lncRNA function exploration by providing some references.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

In this thesis, I aim to present machine learning tools for lncRNA functional annotation. The lncRNA functional annotation mainly comprises two parts: lncRNA annotation and lncRNA function identification. In lncRNA annotation, I start by identifying the RNA transcripts from background DNA sites, then I try to distinguish the lncRNA from mRNA. After that, I develop computational approaches to indicate the lncRNA functions by identifying the types of biomolecules that a lncRNA would interact with and predicting the DNA:lncRNA triplexes in practice.

Starting from the transcript annotation, in Chapter 3, I investigate if transcript and exon usage could be predicted from transcription factor and chromatin interaction-related data. Epigenomics and transcriptomics data from HTS techniques like RNA-seq and ChIP-seq have been successfully applied in predicting alternative splicing, but the locations of chromatin loops in the genome identified by techniques such as ChIA-PET has never been used for prediction tasks. I develop machine learning models to investigate if ChIA-PET could contribute to transcript and exon usage prediction. In doing so, I use a large set of transcription factors as well as ChIA-PET data. I develop different Gradient Boosting Trees models according to the different tasks with the integrated datasets from three cell lines, including GM12878, HeLaS3, and K562. I validate the models via 10-fold

cross-validation, chromosome-split validation, and cross-cell validation. Our results show that both transcript and splicing-derived exon usage can be effectively predicted with at least 0.7512 and 0.7459 of accuracy, respectively, on all cell lines from all kinds of validations. Examining the predictive features, I find that RNA Polymerase II ChIA-PET is one of the most important features in both transcript and exon usage prediction, suggesting that chromatin loop anchors are predictive of both transcript and exon usage.

The question follows the transcription identification is the determination of transcript types. For this purpose, in Chapter 4, I focus on distinguishing lncRNA from mRNA. Previous methods did not perform well on RNAs with sORFs, and they mainly developed models using *human* data. To overcome the limitations of current methods, I develop DeepCPP. I pay particular attention to the RNAs with sORF, and I construct models for *human*, vertebrate, and insect species. I propose a novel feature to extract the nucleotide information around the start codon, named the nucleotide bias feature. This feature is demonstrated to be practical and useful to classify sORF type RNAs which is shown to achieve better performance than other feature representation methods. I evaluate DeepCPP using data from different species and compare it to other recently developed approaches. Results show that DeepCPP outperforms other tools in most cases. Particularly, it improves the identification accuracy of mRNA with sORF significantly. The comprehensive comparisons on different test datasets demonstrate the effectiveness of DeepCPP in distinguishing lncRNA from mRNA.

After the lncRNA annotation, the next two chapters discuss lncRNA function prediction. lncRNAs mainly execute their functions via the interaction with DNA, RNA, and protein. Therefore, in Chapter 5, I propose lncIBTP to indicate the possible interaction biomolecule types for a lncRNA. Because I believe that identifying the types of interaction biomolecules for a lncRNA as a pre-step could improve the accuracy and efficiency for those specific lncRNA-biomolecule prediction approaches. lncIBTP performs well on identifying the types of biomolecules in the interactions with lncRNA, which may contribute to the investigation of functions and characteristics for lncRNAs. But according to our collected data and the prediction results, I find that lncRNA is less likely to interact with RNA and DNA at the same time without interacting with protein.

In Chapter 6, I focus on a specific type of interaction of lncRNA to decipher the code of lncRNA function, that is the DNA:lncRNA triplex-formation. I present a deep

learning program called TriplexFPP to overcome the limitations of current computational methods which identify lots of triplex-forming lncRNAs while lacking any features from experimentally verified data. TriplexFPP predicts the most possible lncRNAs to form triplex from all the theoretically identified triplex-forming lncRNAs by previous canonical rule-based computational approaches, and it indicates the possibility for a DNA site to form a triplex. As the first machine learning tool in DNA:lncRNA triplexes prediction as far as I can tell, TriplexFPP achieves general good performances and it narrows the range of lncRNAs that could form triplex in practice from previous computational approaches.

Therefore, the analysis conducted in this thesis demonstrates some main aspects of lncRNA functional annotation.

Except for Chapter 3 where I use transcription factor and chromatin interaction-related data and GB model, the data and model used in Chapter 4-6 are all sequence data and CNN model. The chosen of GB model in Chapter 3 is based on the nature of the data, that is the data contains lots of zero values. Splitting data to different classes using GB model would be more appropriate than multiplying them by a parameter like what CNN does. The features used in Chapter 5 and 6 are all sequence intrinsic features. Other types of features which are supported by biological evidence to distinguish mRNA and lncRNA are further considered in Chapter 4 and a feature selection method is applied to reduce the model complexity. The architectures and parameters of the models in different prediction tasks with different feature sets are therefore determined correspondingly, where Chapter 4 and 6 use two convolution layers and Chapter 5 uses four convolution layers.

7.2 Future Directions

lncRNA has received much attention in recent years but its functional annotation is rather complex. In this thesis, only a few aspects and certain scenarios are covered and discussed. Our future directions mainly include two parts: 1) improve upon current approaches to achieve better prediction performances, and 2) develop novel methods to be applied on other occasions.

7.2.1 Improve upon Current Approaches

The amount of data and its qualities are the keys to the success of machine learning models. The databases like Ensembl and GENCODE are continuously updated as more and more lncRNAs being annotated or corrected. And also, more working mechanisms for lncRNAs are being revealed with the development of the biology techniques. The fast expanded databases provide us foundations to develop more accurate prediction tools. Some limitations caused by the data amount or quality in our current tools could be overcome to some degree. For example, the prediction of lncRNAs that only interacting with RNA in lncIBTP could be improved as more such kinds of data being explored, and the performance of TriplexFPP could be more persuasive by correcting the lncRNAs that could form triplex with DNA from the current negative datasets.

Besides, except for the transcript and exon usage prediction, all other prediction tools developed in this thesis use sequence data as the raw data. To be used as the entry of machine learning models, the raw sequences need to be converted to numerical representations. Currently, I use feature extraction strategies like k-mer to extract information from the raw sequences. However, these features cannot fully represent the original sequence information. Therefore, more feature extraction algorithms should be explored and the deep learning-based feature extractor may be used to mine as much information from the raw sequences.

Furthermore, lots of novel machine learning structures are proposed every year, but most of them are initially and mainly applied to the area of the image or natural language processing. If these novel models could be applied and improved on lncRNA functional annotation also remains to be explored.

7.2.2 New Application Opportunities

7.2.3 lncRNA Identification

In the first stage of lncRNA function annotation, *i.e.* lncRNA annotation, I only develop the method to distinguish lncRNA from mRNA after identifying the transcripts from background sites. There are many other kinds of RNAs like circular ncRNAs (circRNAs),

ribosomal RNAs (rRNAs) and transfer RNAs (tRNAs) (Fig.7.1) [193,194]. circRNAs are recent and new-growing components of the lncRNAs class [195]. circRNAs are a kind of single-stranded RNA that form covalently closed continuous loops. But circRNA is usually considered as a different RNA group from linear lncRNA (referred to lncRNA in this thesis) [194]. rRNA and tRNA are housekeeping RNAs and they take part in protein synthesis [196]. Although circRNA and housekeeping RNAs can be easily distinguished from lncRNA from the structural or functions, it is costly and time-intensive to conduct the corresponding biology experiments. Specifically, these RNAs usually have comparable sequence lengths with the lncRNA, which makes them impossible to be distinguished from lncRNA like those small ncRNAs according to the sequence lengths.

Some tools are available to make a classification between lncRNA and circRNA. For example, an end-to-end machine learning framework named circDeep is proposed for robust circRNA prediction by introducing different feature descriptors [197], CirRNAPL adopts the idea of particle swarm optimization algorithm to develop an optimized extreme learning machine approach [198], and circLGB is presented considering both sequence information and three new features containing the internal ribosome entry site and the information related to adenosine to inosine [199]. If I can integrate the information and develop a one-step *in silico* tool to distinguish lncRNA from other types of RNAs using sequences after being identified as transcripts, the efficiencies for lncRNA annotation could be largely improved.

7.2.4 lncRNA Function Prediction

It is reported that lncRNAs execute their functions mainly through the interaction with other biomolecules [25]. Therefore, predicting the interactions between lncRNAs and other molecules is considered to be essential in lncRNA function prediction. The major types of lncRNA-molecule interactions are summarized in Chapter 5. I have developed lncIBTP to predict the possible interactions of lncRNA, but more specific tools are needed to find which biomolecules or sites a lncRNA could interact with.

Some tools have been developed to find if a lncRNA is associated with a certain biomolecule or sites as I discussed in Chapter 2. For example, lncRNA-miRNA prediction tools like PmliPred [72] and CIRNN [73]; lncRNA-protein prediction tools like

Fig. 7.1: An illustration of major groups of RNA. RNAs mainly can be divided into non-coding RNAs (ncRNAs) and messenger RNAs (mRNAs). ncRNAs consist of linear ncRNAs and circular RNAs (circRNAs). Linear ncRNAs consist of housekeeping ncRNAs and regulatory ncRNAs. Housekeeping ncRNAs consist of transfer RNAs (tRNAs), ribosomal RNAs (rRNAs), small nucleolar RNAs (snoRNAs), and small nuclear RNAs (snRNAs). Regulatory ncRNAs comprise of small ncRNAs and long non-coding RNAs (lncRNAs). Small ncRNAs contain PIWI-interacting RNAs (piRNAs), microRNAs (miRNAs), and small interfering RNAs (siRNAs).

HLPI-Ensemble [74], SFPEL-LPI [75], and PLRPIM [76]; lncRNA-DNA prediction tools like DeepLGP [77], Triplexator [79], LongTarget [80], Triplex Domain Finder (TDF) [81], and Triplex-Inspector [82]. And some tools are developed to predict even more complex interaction mechanism, *i.e.*, lncRNA-miRNA-mRNA competing endogenous RNA network. For example, Ye *et al.* combine bioinformatics approach and multi-level expression data and to reveal the mechanisms in lncRNA-miRNA-mRNA regulation [200].

However, there are two main concerns. For one thing, most *in silico* tools are developed to predict lncRNA-miRNA and lncRNA-protein, and some tools are proposed to reveal lncRNA-DNA. As I discussed in Chapter 5, lncRNA also can interact with other ncRNAs or form R-Loop with DNA. However, according to the best of our knowledge, no *in silico* tools are currently available to predict such interactions due to the limitation of data. For another, the available tools use a variety of input data formats, which makes it difficult to follow a uniform pipeline to identify the function for a newly annotated lncRNA. Therefore, in the future, the minor types of lncRNA interaction may be explored when taking full consideration of the data. Besides, it is possible to construct an integrated pipeline for lncRNA functional annotation following the procedure of i) identifying the possible interaction biomolecule types, and ii) identifying the specific interaction biomolecules without re-preparing the input data.

Appendices

Appendix A

Model Determination and Parameter Tuning in Chapter 3

A.1 Parameter Determination for transcription prediction

This section showed the candidate and determined parameters for GB model in different prediction tasks and different cell lines (Fig.A.1, Fig.A.2).

Task	Cell line	Candidate parameters	Determined parameters
Verified transcript prediction	GM12878	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=7 max_depth=5 min_samples_split=50 n_estimators=100 learning_rate=0.15 subsample=0.9
	HeLaS3	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=5 max_depth=5 min_samples_split=10 n_estimators=100 learning_rate=0.15 subsample=0.9
	K562	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=3 max_depth=5 min_samples_split=30 n_estimators=100 learning_rate=0.15 subsample=1
Transcript abundance prediction	GM12878	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=7 max_depth=7 n_estimators=100 learning_rate=0.1 subsample=0.5
	HeLaS3	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=3 max_depth=7 n_estimators=100 learning_rate=0.075 subsample=0.8
	K562	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples leaf=3 max_depth=7 n_estimators=500 learning_rate=0.01 subsample=0.5

Fig. A.1: The candidate and determined parameters for GB model in different transcription prediction task. The parameters were searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds.

Cell line	Candidate parameters	Determined parameters
GM12878	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=3 max_depth=7 min_samples_split=50 n_estimators=500 learning_rate=0.2 subsample=0.95
HeLaS3	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=7 max_depth=7 min_samples_split=50 n_estimators=500 learning_rate=0.2 subsample=0.9
K562	min_samples_leaf = {3,5,7,9} max_depth = {4,5,6,7} n_estimators = {100,200,500} min_samples_split = {4, 10,20,30,40,50,60} learning_rate = {0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2} subsample = {[0.5, 0.8, 0.9, 0.95, 1.0]}	min_samples_leaf=7 max_depth=7 min_samples_split=40 n_estimators=500 learning_rate=0.2 subsample=0.9

Fig. A.2: The candidate and determined parameters for GB model in different exon usage prediction task. The parameters were searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds..

Appendix B

Model Determination and Parameter Tuning in Chapter 4

B.1 Comparisons among Different Machine Learning Models

We compare different machine learning models using humna training data with 10-fold validation, the results were shown in Fig.B.1. The parameters used for SVM, RF, and GB were the ones that achieving the best results, searched by *GridSearchCV* function in python package *sklearn.model_selection* with 5 folds (Fig.B.2). The parameters used for CNN model are two convulutional layers with filter number, kernel size, and stride equal to 20, 2, and 1, followed by two dense layers whose dense sizes are 400 and 2, and 'relu' activation functions. The parameters used for NN model were the same as that of the CNN but removing the two convolutional layers. We chose to use CNN model as our final model which achieved best results.

Fig. B.1: The comparison among different machine learning models using training data with 10-fold validation.

Model	Candidate parameters	Determined parameters
SVM	$C = \{2, 5, 10, 15, 20, 25, 30\}$ $\gamma = \{0.1, 0.2, 0.5, 1, 10, 'scale'\}$	$\text{kernel}='rbf'$ $C=30$ $\gamma='scale'$
RF	$\text{min_samples_leaf} = \{3, 5, 7, 9\}$ $\text{max_depth} = \{4, 5, 6, 7\}$ $\text{min_samples_split} = \{2, 4, 6, 8, 10, 20, 40, 60\}$ $\text{n_estimators} = \{100, 200, 300, 400, 500\}$	$\text{min_samples_leaf}=3$ $\text{max_depth}=7$ $\text{min_samples_split}=20$ $\text{n_estimators}=500$ $\text{max_features}='sqrt'$
GB	$\text{min_samples_leaf} = \{3, 5, 7, 9\}$ $\text{max_depth} = \{4, 5, 6, 7\}$ $\text{min_samples_split} = \{2, 4, 6, 8, 10, 20, 40, 60\}$ $\text{n_estimators} = \{100, 200, 300, 400, 500\}$ $\text{subsample} = \{0.7, 0.8, 0.9, 1\}$ $\text{learning_rate} = \{0.1, 0.05, 0.01, 0.005, 0.001\}$	$\text{min_samples_leaf}=7$ $\text{max_depth}=6$ $\text{min_samples_split}=20$ $\text{n_estimators}=200$ $\text{subsample}=0.8$ $\text{learning_rate}=0.001$ $\text{max_features}='sqrt'$

Fig. B.2: The candidate and determined parameters for SVM, RF, and GB model. The parameters were searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds.

B.2 Network Parameter Determination

The model architecture and parameters for human normal type model was determined using our human training data. We randomly split the data to train and test with ratio of 8:2. We tested and determined the values of parameter according to the best Acc on test data with several choices for each parameter (Fig.B.3). The final normal type model was determined as two CNN layers, whose filter number, kernel size, and stride were 20, 8, and 1, and 20, 2, and 1; the dropout rate was selected as 0.2; the dense layer size was determined as 300 and the activation function was 'relu'; trained with 105 epoches.

The model architecture and parameters for human sORF type model was determined by training with our human training data and testing on CPPred human sORF test data ($D_{test-H-sORF}$). We tested and determined the values of parameter according to the best Acc on test data (Fig.B.4). The final sORF type model was determined as two CNN layers, whose filter number, kernel size, and stride were 60, 10, and 4, and 80, 4, and 1; the dropout rate was selected as 0.2; the dense layer size was determined as 400 and the activation function was 'relu'; trained with 45 epoches.

Fig. B.3: DeepCPP normal prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions in dense layer, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. dense size of dense layer, and h. training epochs (the best test accuracy is marked by dash line).

Fig. B.4: DeepCPP sORF prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions in dense layer, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. dense size of dense layer, and h. training epochs (the best test accuracy is marked by dash line).

Appendix C

Model Determination and Parameter Tuning in Chapter 5

C.1 Network Parameter Determination

We randomly split each class to training and test with ratio of 8:2. We over-sampled or downsampled each class of training data to sample number of samples. We tested and determined the values of parameter according to the best Acc on test data with several choices for each parameter (Fig.C.1). The number of balanced training samples for each class was determined as 2000. The final model was determined as four CNN layers, whose filter number was 40 for the first layer and 20 for the remaining layers, and kernel size and stride were all set as 2 and 1; the dropout rate was selected as 0.2; the dense layer size was determined as 400 and the activation function was 'relu'; the optimizer was chose as 'adam'; trained with 80 epoches.

C.2 Determination of Ensemble

Our results show that ensembling different models could improve the performances. We chose to ensemble 5 models which achieved the best performance on the smallest class (lncRNAs interacting with DNA-related molecules) according to Fig.C.2.

Fig. C.1: Network parameter determination. Accuracies achieved with different choices of a. sample numbers in each class (The models are trained with 1 convolutional layer (filter number=40, kernel size=2, strides=1), 50 epochs, and 20% dropout), b. numbers of convolution layers, c. dropout rates, d. activation functions, e. kernel size in the 1st convolution layer, f. optimizers, g dense layer sizes, and h. training epochs.

Number of ensemble models	AUC values of the interaction class of RNA				AUC values of the interaction class of DNA- related			
	3	5	7	10	3	5	7	10
Fold 1	0.7004	0.7376	0.7382	0.7407	0.8611	0.8576	0.8542	0.8513
Fold 2	0.6744	0.7207	0.7216	0.7225	0.8198	0.8182	0.8280	0.8210
Fold 3	0.6796	0.7000	0.7074	0.7065	0.8291	0.8542	0.8505	0.8613
Fold 4	0.7935	0.7872	0.7815	0.7802	0.8373	0.8443	0.8414	0.8414
Fold 5	0.6783	0.6715	0.6709	0.6703	0.8002	0.7967	0.7934	0.7922
Average	0.7052	0.7234	0.7239	0.7240	0.8295	0.8342	0.8335	0.8334

Fig. C.2: The performances with different number of ensemble models.

C.3 Parameter Determination for Other Models

This section showed the candidate and determined parameters of baseline models used for comparison, where the parameters were determined as the ones that achieving the optimal accuracy, searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds.

Model	Candidate parameters	Determined parameters
SVM	C = {1,2,3,4,5,6,7,8,9,10,20,30,40,50}	kernel='rbf'
	Gamma = {0.001, 0.01, 0.1, 1, 'scale'}	C=30
		gamma='scale'
RF	min_samples_leaf = {3,5,7,9}	min_samples_leaf=5
	max_depth = {4,5,6,7}	max_depth=7
	min_samples_split = {4,10,20,60}	min_samples_split=4
	n_estimators = {100,250,500}	n_estimators=500
	min_samples_split = {2,4,6,8,10,20,40,60}	min_samples_split=4
KNN	leaf_size = {range (1,50)}	leaf_size=1
	n_neighbors = {range (1,50)}	n_neighbors=1

Fig. C.3: The candidate and determined parameters for baseline models. The parameters were searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds.

Appendix D

Model Determination and Parameter Tuning in Chapter 6

D.1 Network Parameter Determination

We randomly split the dataset to training and test with ratio of 8:2. We tested and determined the values of parameter according to the optimal value of sensitivity on test data with several choices for each parameter for triplex lncRNA prediction (Fig.D.1) and triplex DNA prediction separately (Fig.D.2). The final triplex lncRNA prediction model was determined as two CNN layers, whose filter number, kernel size and stride were all set as 20, 2, and 1; the dropout rate was selected as 0.2; the dense layer size was determined as 400 and the activation function was 'relu'; the optimizer was chose as 'adam'; trained with 110 epoches. The final triplex DNA prediction model was determined as two CNN layers, whose filter number, kernel size and stride were 20, 8, and 2, and 20, 2, and 1; the dropout rate was selected as 0.2; the dense layer size was determined as 400 and the activation function was 'relu'; the optimizer was chose as 'adam'; trained with 75 epoches.

Fig. D.1: TriplexFPP triplex lncRNA prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. optimizers, and h. training epochs (the best test accuracy is marked by dash line).

Fig. D.2: TriplexFPP triplex DNA site prediction model network parameter determination. Accuracies achieved with different choices of a. numbers of convolution layers, b. activation functions, c. kernel and stride size in the 1st convolution layer, d. kernel and stride size in the 2nd convolution layer, e. filter number in the 1st and 2nd convolution layer, f. dropout rates, g. optimizers, and h. training epochs (the best test accuracy is marked by dash line).

D.2 Parameter Determination for Other Models

This section showed the candidate and determined parameters of baseline models used for comparison, where the parameters were determined as the ones that achieving the optimal recall (or sensitivity), searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds.

Model	Candidate parameters	Determined parameters
SVM	C = {2, 5, 10, 15, 20, 25, 30}	kernel='rbf'
	gamma = {0.1, 0.2, 0.5, 1, 10, 'scale'}	C=25 gamma='scale'
RF	min_samples_leaf = {3, 5, 7, 9}	min_samples_leaf=3
	max_depth = {4, 5, 6, 7}	max_depth=7
	min_samples_split = {2, 4, 6, 8, 10, 20, 40, 60}	min_samples_split=2
	n_estimators = {100, 200, 300, 400, 500}	n_estimators=200 max_features='sqrt'
GB	min_samples_leaf = {3, 5, 7, 9}	min_samples_leaf=9
	max_depth = {4, 5, 6, 7}	max_depth=5
	min_samples_split = {2, 4, 6, 8, 10, 20, 40, 60}	min_samples_split=4
	n_estimators = {100, 200, 300, 400, 500}	n_estimators=200
	subsample = {0.7, 0.8, 0.9, 1}	subsample=1
	learning_rate = {0.1, 0.05, 0.01, 0.005, 0.001}	learning_rate=0.1 max_features='sqrt'
NN	Number_of_layer = {1, 2, 3}	Number_of_layer = 2
	Dense_size = {16, 32, 64, 256}	Layer1_dense_size = 64 Layer2_dense_size = 16
	Activation_function = {relu, tanh, sigmoid, softmax}	Activation_function = relu
	Dropout_rate = {0.2, 0.3, 0.4, 0.5}	Dropout_rate = 0.2
	Optimizer = {adam, sgd, adadelata, adagrad, adamax}	Optimizer = adam

Fig. D.3: The candidate and determined parameters for baseline models in predicting triplex lncRNA. The parameters for SVM, RF, and GB models are searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds and 'recall' scoring method.

Model	Candidate parameters	Determined parameters
SVM	C = {2, 5, 10, 15, 20, 25, 30}	kernel='rbf'
	gamma = {0.1, 0.2, 0.5, 1, 10, 'scale'}	C=2 gamma=1
RF	min_samples_leaf = {3, 5, 7, 9}	min_samples_leaf=7
	max_depth = {4, 5, 6, 7}	max_depth=7
	min_samples_split = {2, 4, 6, 8, 10, 20, 40, 60}	min_samples_split=2
	n_estimators = {100, 200, 300, 400, 500}	n_estimators=500 max_features='sqrt'
GB	min_samples_leaf = {3, 5, 7, 9}	min_samples_leaf=9
	max_depth = {4, 5, 6, 7}	max_depth=4
	min_samples_split = {2, 4, 6, 8, 10, 20, 40, 60}	min_samples_split=60
	n_estimators = {100, 200, 300, 400, 500}	n_estimators=100
	subsample = {0.7, 0.8, 0.9, 1}	subsample=0.9
	learning_rate = {0.1, 0.05, 0.01, 0.005, 0.001}	learning_rate=0.05 max_features='sqrt'
NN	Number_of_layer = {1, 2, 3}	Number_of_layer = 2
	Dense_size = {16, 32, 64, 256}	Layer1_dense_size = 64 Layer2_dense_size = 16
	Activation_function = {relu, tanh, sigmoid, softmax}	Activation_function = relu
	Dropout_rate = {0.2, 0.3, 0.4, 0.5}	Dropout_rate = 0.2
	Optimizer = {adam, sgd, adadelta, adagrad, adamax}	Optimizer = adam

Fig. D.4: The candidate and determined parameters for baseline models in predicting triplex DNA sites. The parameters for SVM, RF, and GB models are searched by *sklearn.model_selection.GridSearchCV* function in python with 5 folds and 'recall' scoring method.

References

- [1] S. Clancy and W. Brown, “Translation: Dna to mrna to protein,” *Nature Education*, vol. 1, no. 1, p. 101, 2008.
- [2] S. Clancy *et al.*, “Rna splicing: introns, exons and spliceosome,” *Nature Education*, vol. 1, no. 1, p. 31, 2008.
- [3] Y. Zheng, “Chapter 1 - introduction to non-coding rnas and high throughput sequencing,” in *Computational Non-coding RNA Biology*, Y. Zheng, Ed. Academic Press, 2019, pp. 3–31. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128143650000105>
- [4] X.-Z. Zhang, H. Liu, and S.-R. Chen, “Mechanisms of long non-coding rnas in cancers and their dynamic regulations,” *Cancers*, vol. 12, no. 5, p. 1245, 2020.
- [5] S. G. Gray, “Chapter 17 - the potential of epigenetic compounds in treating diabetes,” in *Epigenetics in Human Disease (Second Edition)*, second edition ed., ser. Translational Epigenetics, T. O. Tollefsbol, Ed. Academic Press, 2018, vol. 6, pp. 489–547. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128122150000170>
- [6] G. Latgé, C. Poulet, V. Bours, C. Josse, and G. Jerusalem, “Natural antisense transcripts: molecular mechanisms and implications in breast cancers,” *International journal of molecular sciences*, vol. 19, no. 1, p. 123, 2018.
- [7] T. R. Mercer and J. S. Mattick, “Structure and function of long noncoding rnas in epigenetic regulation,” *Nature structural & molecular biology*, vol. 20, no. 3, pp. 300–307, 2013.

REFERENCES

- [8] M. Guttman and J. L. Rinn, “Modular regulatory principles of large non-coding rnas,” *Nature*, vol. 482, no. 7385, pp. 339–346, 2012.
- [9] J. M. Engreitz, N. Ollikainen, and M. Guttman, “Long non-coding rnas: spatial amplifiers that control nuclear structure and gene expression,” *Nature reviews Molecular cell biology*, vol. 17, no. 12, pp. 756–770, 2016.
- [10] F. Ferre, A. Colantoni, and M. Helmer-Citterich, “Revealing protein–lncrna interaction,” *Briefings in bioinformatics*, vol. 17, no. 1, pp. 106–116, 2016.
- [11] X. Chen, C. C. Yan, X. Zhang, and Z.-H. You, “Long non-coding rnas and complex diseases: from experimental results to computational models,” *Briefings in bioinformatics*, vol. 18, no. 4, pp. 558–576, 2017.
- [12] Y. Zhang, Y. Tao, and Q. Liao, “Long noncoding rna: a crosslink in biological regulatory network,” *Briefings in bioinformatics*, vol. 19, no. 5, pp. 930–945, 2018.
- [13] S. Yotsukura, D. duVerle, T. Hancock, Y. Natsume-Kitatani, and H. Mamitsuka, “Computational recognition for long non-coding rna (lncrna): software and databases,” *Briefings in bioinformatics*, vol. 18, no. 1, pp. 9–27, 2017.
- [14] F. P. Marchese, I. Raimondi, and M. Huarte, “The multidimensional mechanisms of long noncoding rna function,” *Genome biology*, vol. 18, no. 1, pp. 1–13, 2017.
- [15] L. Statello, C.-J. Guo, L.-L. Chen, and M. Huarte, “Gene regulation by long non-coding rnas and its biological functions,” *Nature Reviews Molecular Cell Biology*, pp. 1–23, 2020.
- [16] L. Poliseno, L. Salmena, J. Zhang, B. Carver, W. J. Haveman, and P. P. Pandolfi, “A coding-independent function of gene and pseudogene mrnas regulates tumour biology,” *Nature*, vol. 465, no. 7301, pp. 1033–1038, 2010.
- [17] I. Morán, I. Akerman, M. Van De Bunt, R. Xie, M. Benazra, T. Nammo, L. Arnes, N. Nakić, J. García-Hurtado, S. Rodríguez-Seguí *et al.*, “Human β cell transcriptome analysis uncovers lncrnas that are tissue-specific, dynamically regulated, and abnormally expressed in type 2 diabetes,” *Cell metabolism*, vol. 16, no. 4, pp. 435–448, 2012.

REFERENCES

- [18] M. N. Ziats and O. M. Rennert, “Aberrant expression of long noncoding rnas in autistic brain,” *Journal of molecular neuroscience*, vol. 49, no. 3, pp. 589–593, 2013.
- [19] Y. Sánchez and M. Huarte, “Long non-coding rnas: challenges for diagnosis and therapies,” *Nucleic acid therapeutics*, vol. 23, no. 1, pp. 15–20, 2013.
- [20] J. Zhang, Z. Lin, Y. Gao, and T. Yao, “Downregulation of long noncoding rna meg3 is associated with poor prognosis and promoter hypermethylation in cervical cancer,” *Journal of experimental & clinical cancer research*, vol. 36, no. 1, pp. 1–9, 2017.
- [21] X. Yang, W. Zhang, S. Cheng, and R. Yang, “High expression of lncrna gacat3 inhibits invasion and metastasis of non-small cell lung cancer to enhance the effect of radiotherapy,” *Eur Rev Med Pharmacol Sci*, vol. 22, no. 5, pp. 1315–1322, 2018.
- [22] K. Lou, Z. Li, P. Wang, Z. Liu, Y. Chen, X. Wang, and H. Cui, “Long non-coding rna bancr indicates poor prognosis for breast cancer and promotes cell proliferation and invasion,” *Eur Rev Med Pharmacol Sci*, vol. 22, no. 5, pp. 1358–1365, 2018.
- [23] J. Pan, “Lncrna h19 promotes atherosclerosis by regulating mapk and nf-kb signaling pathway,” *Eur Rev Med Pharmacol Sci*, vol. 21, no. 2, pp. 322–328, 2017.
- [24] J. T. Kung, D. Colognori, and J. T. Lee, “Long noncoding rnas: past, present, and future,” *Genetics*, vol. 193, no. 3, pp. 651–669, 2013.
- [25] Y. Ramakrishnaiah, L. Kuhlmann, and S. Tyagi, “linc2function: A deep learning model to identify and assign function to long noncoding rna (lncrna),” *bioRxiv*, 2021.
- [26] J. A. Reuter, D. V. Spacek, and M. P. Snyder, “High-throughput sequencing technologies,” *Molecular cell*, vol. 58, no. 4, pp. 586–597, 2015.
- [27] D. Singh, P. K. Singh, S. Chaudhary, K. Mehla, and S. Kumar, “Exome sequencing and advances in crop improvement,” *Advances in genetics*, vol. 79, pp. 87–121, 2012.
- [28] D. B. Sawyer and R. S. Vasan, *Encyclopedia of Cardiovascular Research and Medicine*. Elsevier, 2017.

REFERENCES

- [29] I. I. Muhammad, S. L. Kong, S. N. Akmar Abdullah, and U. Munusamy, “Rna-seq and chip-seq as complementary approaches for comprehension of plant transcriptional regulatory mechanism,” *International journal of molecular sciences*, vol. 21, no. 1, p. 167, 2020.
- [30] M. J. Fullwood and Y. Ruan, “Chip-based methods for the identification of long-range chromatin interactions,” *Journal of cellular biochemistry*, vol. 107, no. 1, pp. 30–39, 2009.
- [31] C. Chu, K. Qu, F. L. Zhong, S. E. Artandi, and H. Y. Chang, “Genomic maps of long noncoding rna occupancy reveal principles of rna-chromatin interactions,” *Molecular cell*, vol. 44, no. 4, pp. 667–678, 2011.
- [32] M. D. Simon, C. I. Wang, P. V. Kharchenko, J. A. West, B. A. Chapman, A. A. Alekseyenko, M. L. Borowsky, M. I. Kuroda, and R. E. Kingston, “The genomic binding sites of a noncoding rna,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 51, pp. 20 497–20 502, 2011.
- [33] J. M. Engreitz, A. Pandya-Jones, P. McDonel, A. Shishkin, K. Sirokman, C. Surka, S. Kadri, J. Xing, A. Goren, E. S. Lander *et al.*, “The xist lncrna exploits three-dimensional genome architecture to spread across the x chromosome,” *Science*, vol. 341, no. 6147, 2013.
- [34] T. Mondal, S. Subhash, R. Vaid, S. Enroth, S. Uday, B. Reinius, S. Mitra, A. Mohammed, A. R. James, E. Hoberg *et al.*, “Meg3 long noncoding rna regulates the $\text{tgf-}\beta$ pathway genes through formation of rna–dna triplex structures,” *Nature communications*, vol. 6, no. 1, pp. 1–17, 2015.
- [35] H.-L. V. Wang and J. A. Chekanova, “An overview of methodologies in studying lncnas in the high-throughput era: When acronyms attack!” *Plant Long Non-Coding RNAs*, pp. 1–30, 2019.
- [36] A. Bassett, A. Akhtar, D. Barlow, A. Bird, N. Brockdorff, D. Duboule, A. Ephrussi, A. Ferguson-Smith, T. Gingeras, W. Haerty *et al.*, “Considerations when investigating lncrna function in vivo. *elife* 3: e03058,” 2014.

REFERENCES

- [37] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko *et al.*, “Ncbi geo: archive for functional genomics data sets—update,” *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2012.
- [38] K. Pruitt, G. Brown, T. Tatusova, D. Maglott, and J. McEntyre, “Chapter 18: The reference sequence (refseq) database,” *The NCBI Handbook [Internet]*, 2002.
- [39] N. A. O’Leary, M. W. Wright, J. R. Brister, S. Ciufu, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei *et al.*, “Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation,” *Nucleic acids research*, vol. 44, no. D1, pp. D733–D745, 2016.
- [40] J. Navarro Gonzalez, A. S. Zweig, M. L. Speir, D. Schmelter, K. R. Rosenbloom, B. J. Raney, C. C. Powell, L. R. Nassar, N. D. Maulding, C. M. Lee *et al.*, “The ucsc genome browser database: 2021 update,” *Nucleic Acids Research*, vol. 49, no. D1, pp. D1046–D1057, 2021.
- [41] C. A. Davis, B. C. Hitz, C. A. Sloan, E. T. Chan, J. M. Davidson, I. Gabdank, J. A. Hilton, K. Jain, U. K. Baymuradov, A. K. Narayanan *et al.*, “The encyclopedia of dna elements (encode): data portal update,” *Nucleic acids research*, vol. 46, no. D1, pp. D794–D801, 2018.
- [42] J. E. Moore, M. J. Purcaro, H. E. Pratt, C. B. Epstein, N. Shores, J. Adrian, T. Kawli, C. A. Davis, A. Dobin, R. Kaul *et al.*, “Expanded encyclopaedias of dna elements in the human and mouse genomes,” *Nature*, vol. 583, no. 7818, pp. 699–710, 2020.
- [43] A. Frankish, M. Diekhans, I. Jungreis, J. Lagarde, J. E. Loveland, J. M. Mudge, C. Sisu, J. C. Wright, J. Armstrong, I. Barnes *et al.*, “Genome 2021,” *Nucleic Acids Research*, vol. 49, no. D1, pp. D916–D923, 2021.
- [44] S. E. Hunt, W. McLaren, L. Gil, A. Thormann, H. Schuilenburg, D. Sheppard, A. Parton, I. M. Armean, S. J. Trevanion, P. Flicek *et al.*, “Ensembl variation resources,” *Database*, vol. 2018, 2018.

REFERENCES

- [45] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code,” *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.
- [46] A. Jha, M. R. Gazzara, and Y. Barash, “Integrative deep models for alternative splicing,” *Bioinformatics*, vol. 33, no. 14, pp. i274–i282, 2017.
- [47] Z. Zhang, Z. Pan, Y. Ying, Z. Xie, S. Adhikari, J. Phillips, R. P. Carstens, D. L. Black, Y. Wu, and Y. Xing, “Deep-learning augmented rna-seq analysis of transcript splicing,” *Nature methods*, vol. 16, no. 4, pp. 307–310, 2019.
- [48] D. Lee, J. Zhang, J. Liu, and M. Gerstein, “Epigenome-based splicing prediction using a recurrent neural network,” *PLoS computational biology*, vol. 16, no. 6, p. e1008006, 2020.
- [49] I. Ulitsky and D. P. Bartel, “lincrnas: genomics, evolution, and mechanisms,” *Cell*, vol. 154, no. 1, pp. 26–46, 2013.
- [50] L. Kong, Y. Zhang, Z.-Q. Ye, X.-Q. Liu, S.-Q. Zhao, L. Wei, and G. Gao, “Cpc: assess the protein-coding potential of transcripts using sequence features and support vector machine,” *Nucleic acids research*, vol. 35, no. suppl_2, pp. W345–W349, 2007.
- [51] M. F. Lin, I. Jungreis, and M. Kellis, “PhyloCSF: a comparative genomics method to distinguish protein coding and non-coding regions,” *Bioinformatics*, vol. 27, no. 13, pp. i275–i282, 2011.
- [52] L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher, and W. Li, “Cpat: Coding-potential assessment tool using an alignment-free logistic regression model,” *Nucleic acids research*, vol. 41, no. 6, pp. e74–e74, 2013.
- [53] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, and Y. Zhao, “Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts,” *Nucleic acids research*, vol. 41, no. 17, pp. e166–e166, 2013.
- [54] A. Li, J. Zhang, and Z. Zhou, “Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme,” *BMC bioinformatics*, vol. 15, no. 1, pp. 1–10, 2014.

REFERENCES

- [55] L. Sun, H. Liu, L. Zhang, and J. Meng, “Incrscan-svm: a tool for predicting long non-coding rnas using support vector machine,” *PloS one*, vol. 10, no. 10, p. e0139654, 2015.
- [56] R. Achawanantakun, J. Chen, Y. Sun, and Y. Zhang, “Lncrna-id: Long non-coding rna identification using balanced random forests,” *Bioinformatics*, vol. 31, no. 24, pp. 3897–3905, 2015.
- [57] Y.-J. Kang, D.-C. Yang, L. Kong, M. Hou, Y.-Q. Meng, L. Wei, and G. Gao, “Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features,” *Nucleic acids research*, vol. 45, no. W1, pp. W12–W16, 2017.
- [58] L. Hu, Z. Xu, B. Hu, and Z. J. Lu, “Come: a robust coding potential calculation tool for lncrna identification and characterization based on multiple features,” *Nucleic acids research*, vol. 45, no. 1, pp. e2–e2, 2017.
- [59] V. Wucher, F. Legeai, B. Hedan, G. Rizk, L. Lagoutte, T. Leeb, V. Jagannathan, E. Cadieu, A. David, H. Lohi *et al.*, “Feelnc: a tool for long non-coding rna annotation and its application to the dog transcriptome,” *Nucleic acids research*, vol. 45, no. 8, pp. e57–e57, 2017.
- [60] H. W. Schneider, T. Raiol, M. M. Brigido, M. E. M. Walter, and P. F. Stadler, “A support vector machine based method to distinguish long non-coding rnas from protein coding transcripts,” *BMC genomics*, vol. 18, no. 1, pp. 1–14, 2017.
- [61] S. Han, Y. Liang, Q. Ma, Y. Xu, Y. Zhang, W. Du, C. Wang, and Y. Li, “Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property,” *Briefings in bioinformatics*, vol. 20, no. 6, pp. 2009–2027, 2019.
- [62] X. Tong and S. Liu, “Cppred: coding potential prediction based on the global description of rna sequence,” *Nucleic acids research*, vol. 47, no. 8, pp. e43–e43, 2019.

REFERENCES

- [63] R. Tripathi, S. Patel, V. Kumari, P. Chakraborty, and P. K. Varadwaj, “Deeplnc, a long non-coding rna prediction tool using deep neural network,” *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 5, no. 1, pp. 1–14, 2016.
- [64] A. P. Camargo, V. Sourkov, and M. F. Carazzolle, “Rnasamba: coding potential assessment using orf and whole transcript sequence information,” *BioRxiv*, p. 620880, 2019.
- [65] S. T. Hill, R. Kuintzle, A. Teegarden, E. Merrill III, P. Danaee, and D. A. Hendrix, “A deep recurrent neural network discovers complex biological rules to decipher rna protein-coding potential,” *Nucleic acids research*, vol. 46, no. 16, pp. 8105–8113, 2018.
- [66] C. Yang, L. Yang, M. Zhou, H. Xie, C. Zhang, M. D. Wang, and H. Zhu, “Lncadeep: an ab initio lncrna identification and functional annotation tool based on deep learning,” *Bioinformatics*, vol. 34, no. 22, pp. 3825–3834, 2018.
- [67] J. Baek, B. Lee, S. Kwon, and S. Yoon, “Lncrnnet: long non-coding rna identification using deep learning,” *Bioinformatics*, vol. 34, no. 22, pp. 3889–3897, 2018.
- [68] H. K. Lee, A. K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis, “Coexpression analysis of human genes across many microarray data sets,” *Genome research*, vol. 14, no. 6, pp. 1085–1094, 2004.
- [69] Q. Jiang, R. Ma, J. Wang, X. Wu, S. Jin, J. Peng, R. Tan, T. Zhang, Y. Li, and Y. Wang, “Lncrna2function: a comprehensive resource for functional investigation of human lncrnas based on rna-seq data,” in *BMC genomics*, vol. 16, no. 3. BioMed Central, 2015, pp. 1–11.
- [70] Z. Zhang, J. Zhang, C. Fan, Y. Tang, and L. Deng, “Katzlgo: large-scale prediction of lncrna functions by using the katz measure based on multiple networks,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 2, pp. 407–416, 2017.

REFERENCES

- [71] J. Zhou, Y. Huang, Y. Ding, J. Yuan, H. Wang, and H. Sun, “lncfunkt: a toolkit for functional annotation of long noncoding rnas,” *Bioinformatics*, vol. 34, no. 19, pp. 3415–3416, 2018.
- [72] Q. Kang, J. Meng, J. Cui, Y. Luan, and M. Chen, “Pmlipred: a method based on hybrid model and fuzzy decision for plant mirna–lncrna interaction prediction,” *Bioinformatics*, vol. 36, no. 10, pp. 2986–2992, 2020.
- [73] P. Zhang, J. Meng, Y. Luan, and C. Liu, “Plant mirna–lncrna interaction prediction with the ensemble of cnn and indrnn,” *Interdisciplinary Sciences: Computational Life Sciences*, vol. 12, no. 1, pp. 82–89, 2020.
- [74] H. Hu, L. Zhang, H. Ai, H. Zhang, Y. Fan, Q. Zhao, and H. Liu, “Hlpi-ensemble: prediction of human lncrna-protein interactions based on ensemble strategy,” *RNA biology*, vol. 15, no. 6, pp. 797–806, 2018.
- [75] W. Zhang, X. Yue, G. Tang, W. Wu, F. Huang, and X. Zhang, “Sfpel-lpi: sequence-based feature projection ensemble learning for predicting lncrna-protein interactions,” *PLoS computational biology*, vol. 14, no. 12, p. e1006616, 2018.
- [76] J. S. Wekesa, Y. Luan, M. Chen, and J. Meng, “A hybrid prediction method for plant lncrna-protein interaction,” *Cells*, vol. 8, no. 6, p. 521, 2019.
- [77] T. Zhao, Y. Hu, J. Peng, and L. Cheng, “Deeplgp: a novel deep learning method for prioritizing lncrna target genes,” *Bioinformatics*, vol. 36, no. 16, pp. 4466–4472, 2020.
- [78] F. Wang, P. Chainani, T. White, J. Yang, Y. Liu, and B. Soibam, “Deep learning identifies genome-wide dna binding sites of long noncoding rnas,” *RNA biology*, vol. 15, no. 12, pp. 1468–1476, 2018.
- [79] F. A. Buske, D. C. Bauer, J. S. Mattick, and T. L. Bailey, “Triplexator: detecting nucleic acid triple helices in genomic and transcriptomic data,” *Genome research*, vol. 22, no. 7, pp. 1372–1381, 2012.

- [80] S. He, H. Zhang, H. Liu, and H. Zhu, “Longtarget: a tool to predict lncrna dna-binding motifs and binding sites via hoogsteen base-pairing analysis,” *Bioinformatics*, vol. 31, no. 2, pp. 178–186, 2015.
- [81] C.-C. Kuo, S. Hänzelmann, N. Sentürk Cetin, S. Frank, B. Zajzon, J.-P. Derks, V. S. Akhade, G. Ahuja, C. Kanduri, I. Grummt *et al.*, “Detection of rna–dna binding sites in long noncoding rnas,” *Nucleic acids research*, vol. 47, no. 6, pp. e32–e32, 2019.
- [82] F. A. Buske, D. C. Bauer, J. S. Mattick, and T. L. Bailey, “Triplex-inspector: an analysis tool for triplex-mediated targeting of genomic loci,” *Bioinformatics*, vol. 29, no. 15, pp. 1895–1897, 2013.
- [83] Y. Zhang, Y. Cai, X. Roca, M. J. Fullwood, and C. K. Kwoh, “Chromatin loop anchors predict transcript and exon usage,” *Briefings in bioinformatics*, 2021.
- [84] S. S. Bjørklund, A. Panda, S. Kumar, M. Seiler, D. Robinson, J. Gheeya, M. Yao, G. I. G. Alnæs, D. Toppmeyer, M. Riis *et al.*, “Widespread alternative exon usage in clinically distinct subtypes of invasive ductal carcinoma,” *Scientific reports*, vol. 7, no. 1, pp. 1–15, 2017.
- [85] A. Reyes, S. Anders, R. J. Weatheritt, T. J. Gibson, L. M. Steinmetz, and W. Huber, “Drift and conservation of differential exon usage across tissues in primate species,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 38, pp. 15 377–15 382, 2013.
- [86] X. Qu, S. Alsager, Y. Zhuo, and B. Shan, “Hox transcript antisense rna (hotair) in cancer,” *Cancer letters*, vol. 454, pp. 90–97, 2019.
- [87] R. A. Eiholzer, S. Mehta, M. Kazantseva, C. J. Drummond, C. McKinney, K. Young, D. Slater, B. C. Morten, K. A. Avery-Kiejda, A. Lasham *et al.*, “Intronic tp53 polymorphisms are associated with increased $\delta 133tp53$ transcript, immune infiltration and cancer risk,” *Cancers*, vol. 12, no. 9, p. 2472, 2020.

REFERENCES

- [88] X. Guo, Q.-R. Chen, Y. K. Song, J. S. Wei, and J. Khan, “Exon array analysis reveals neuroblastoma tumors have distinct alternative splicing patterns according to stage and mycn amplification status,” *BMC medical genomics*, vol. 4, no. 1, pp. 1–11, 2011.
- [89] F. Liu and C.-X. Gong, “Tau exon 10 alternative splicing and tauopathies,” *Molecular neurodegeneration*, vol. 3, no. 1, pp. 1–10, 2008.
- [90] C. Sonesson, M. I. Love, R. Patro, S. Hussain, D. Malhotra, and M. D. Robinson, “A junction coverage compatibility score to quantify the reliability of transcript abundance estimates and annotation catalogs,” *Life science alliance*, vol. 2, no. 1, 2019.
- [91] B. Kakaradov, H. Y. Xiong, L. J. Lee, N. Jojic, and B. J. Frey, “Challenges in estimating percent inclusion of alternatively spliced junctions from rna-seq data,” in *BMC bioinformatics*, vol. 13, no. 6. BioMed Central, 2012, pp. 1–12.
- [92] L. D. Goldstein, Y. Cao, G. Pau, M. Lawrence, T. D. Wu, S. Seshagiri, and R. Gentleman, “Prediction and quantification of splice events from rna-seq data,” *PloS one*, vol. 11, no. 5, p. e0156132, 2016.
- [93] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold, “Genome-wide mapping of in vivo protein-dna interactions,” *Science*, vol. 316, no. 5830, pp. 1497–1502, 2007.
- [94] A. Sekhon, R. Singh, and Y. Qi, “Deepdiff: Deep-learning for predicting differential gene expression from histone modifications,” *Bioinformatics*, vol. 34, no. 17, pp. i891–i900, 2018.
- [95] S. Wang, H. Sun, J. Ma, C. Zang, C. Wang, J. Wang, Q. Tang, C. A. Meyer, Y. Zhang, and X. S. Liu, “Target analysis by integration of transcriptome and chip-seq data with beta,” *Nature protocols*, vol. 8, no. 12, pp. 2502–2515, 2013.
- [96] L.-Q. Zhang, Q.-Z. Li, W.-X. Su, and W. Jin, “Predicting gene expression level by the transcription factor binding signals in human embryonic stem cells,” *Biosystems*, vol. 150, pp. 92–98, 2016.

REFERENCES

- [97] F. Schmidt, N. Gasparoni, G. Gasparoni, K. Gianmoena, C. Cadenas, J. K. Polansky, P. Ebert, K. Nordström, M. Barann, A. Sinha *et al.*, “Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction,” *Nucleic acids research*, vol. 45, no. 1, pp. 54–66, 2017.
- [98] Q. Hu, E. J. Kim, J. Feng, G. R. Grant, and E. A. Heller, “Histone posttranslational modifications predict specific alternative exon subtypes in mammalian brain,” *PLoS computational biology*, vol. 13, no. 6, p. e1005602, 2017.
- [99] M. J. Fullwood, M. H. Liu, Y. F. Pan, J. Liu, H. Xu, Y. B. Mohamed, Y. L. Orlov, S. Velkov, A. Ho, P. H. Mei *et al.*, “An oestrogen-receptor- α -bound human chromatin interactome,” *Nature*, vol. 462, no. 7269, pp. 58–64, 2009.
- [100] J. Dekker and T. Misteli, “Long-range chromatin interactions,” *Cold Spring Harbor perspectives in biology*, vol. 7, no. 10, p. a019356, 2015.
- [101] Y. X. See, B. Z. Wang, and M. J. Fullwood, “Chromatin interactions and regulatory elements in cancer: From bench to bedside,” *Trends in Genetics*, vol. 35, no. 2, pp. 145–158, 2019.
- [102] T. R. Mercer, S. L. Edwards, M. B. Clark, S. J. Neph, H. Wang, A. B. Stergachis, S. John, R. Sandstrom, G. Li, K. S. Sandhu *et al.*, “Dnase i-hypersensitive exons colocalize with promoters and distal regulatory elements,” *Nature genetics*, vol. 45, no. 8, pp. 852–859, 2013.
- [103] Y. Jiang, J. Huang, K. Lun, B. Li, H. Zheng, Y. Li, R. Zhou, W. Duan, C. Wang, Y. Feng *et al.*, “Genome-wide analyses of chromatin interactions after the loss of pol i, pol ii, and pol iii,” *Genome biology*, vol. 21, no. 1, pp. 1–28, 2020.
- [104] F. Cao and M. J. Fullwood, “Inflated performance measures in enhancer–promoter interaction-prediction methods,” *Nature genetics*, vol. 51, no. 8, pp. 1196–1198, 2019.
- [105] F. S. Howe, H. Fischl, S. C. Murray, and J. Mellor, “Is h3k4me3 instructive for transcription activation?” *Bioessays*, vol. 39, no. 1, pp. 1–12, 2017.

REFERENCES

- [106] L. A. Gates, J. Shi, A. D. Rohira, Q. Feng, B. Zhu, M. T. Bedford, C. A. Sagum, S. Y. Jung, J. Qin, M.-J. Tsai *et al.*, “Acetylation on histone h3 lysine 9 mediates a switch from transcription initiation to elongation,” *Journal of Biological Chemistry*, vol. 292, no. 35, pp. 14 456–14 472, 2017.
- [107] E. J. Wagner and P. B. Carpenter, “Understanding the language of lys36 methylation at histone h3,” *Nature reviews Molecular cell biology*, vol. 13, no. 2, pp. 115–126, 2012.
- [108] C. Huang and B. Zhu, “Roles of h3k36-specific histone methyltransferases in transcription: antagonizing silencing and safeguarding transcription fidelity,” *Biophysics reports*, vol. 4, no. 4, pp. 170–177, 2018.
- [109] T. Li, Q. Liu, N. Garza, S. Kornblau, and V. X. Jin, “Integrative analysis reveals functional and regulatory roles of h3k79me2 in mediating alternative splicing,” *Genome medicine*, vol. 10, no. 1, pp. 1–11, 2018.
- [110] J. Soutourina, “Transcription regulation by the mediator complex,” *Nature reviews Molecular cell biology*, vol. 19, no. 4, p. 262, 2018.
- [111] R. Karlić, H.-R. Chung, J. Lasserre, K. Vlahoviček, and M. Vingron, “Histone modification levels are predictive for gene expression,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 7, pp. 2926–2931, 2010.
- [112] X. Dong, M. C. Greven, A. Kundaje, S. Djebali, J. B. Brown, C. Cheng, T. R. Gingeras, M. Gerstein, R. Guigó, E. Birney *et al.*, “Modeling gene expression using chromatin features in various cellular contexts,” *Genome biology*, vol. 13, no. 9, pp. 1–17, 2012.
- [113] O. Kelemen, P. Convertini, Z. Zhang, Y. Wen, M. Shen, M. Falaleeva, and S. Stamm, “Function of alternative splicing,” *Gene*, vol. 514, no. 1, pp. 1–30, 2013.
- [114] H.-L. Zhou, G. Luo, J. A. Wise, and H. Lou, “Regulation of alternative splicing by local histone modifications: potential roles for rna-guided mechanisms,” *Nucleic acids research*, vol. 42, no. 2, pp. 701–713, 2013.

REFERENCES

- [115] I. E. Schor, N. Rascovan, F. Pelisch, M. Alló, and A. R. Kornblihtt, “Neuronal cell depolarization induces intragenic chromatin modifications affecting ncarn alternative splicing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 11, pp. 4325–4330, 2009.
- [116] L. Muniz, E. Nicolas, and D. Trouche, “Rna polymerase ii speed: a key player in controlling and adapting transcriptome composition,” *The EMBO Journal*, vol. 40, no. 15, p. e105740, 2021.
- [117] Q. Sun, Q. Hao, and K. V. Prasanth, “Nuclear long noncoding rnas: key regulators of gene expression,” *Trends in Genetics*, vol. 34, no. 2, pp. 142–157, 2018.
- [118] L. Herzel, D. S. Ottoz, T. Alpert, and K. M. Neugebauer, “Splicing and transcription touch base: co-transcriptional spliceosome assembly and function,” *Nature reviews Molecular cell biology*, vol. 18, no. 10, p. 637, 2017.
- [119] S. Zhang, S. Aibara, S. M. Vos, D. E. Agafonov, R. Lührmann, and P. Cramer, “Structure of a transcribing rna polymerase ii–u1 snrnp complex,” *Science*, vol. 371, no. 6526, pp. 305–309, 2021.
- [120] Y. Zhang, C. Jia, M. J. Fullwood, and C. K. Kwok, “Deepcpp: a deep neural network based on nucleotide bias information and minimum distribution similarity feature selection for rna coding potential prediction,” *Briefings in bioinformatics*, vol. 22, no. 2, pp. 2073–2084, 2021.
- [121] T. R. Mercer, M. E. Dinger, and J. S. Mattick, “Long non-coding rnas: insights into functions,” *Nature reviews genetics*, vol. 10, no. 3, pp. 155–159, 2009.
- [122] S. Djebali, C. A. Davis, A. Merkel, A. Dobin, T. Lassmann, A. Mortazavi, A. Tanzer, J. Lagarde, W. Lin, F. Schlesinger *et al.*, “Landscape of transcription in human cells,” *Nature*, vol. 489, no. 7414, pp. 101–108, 2012.
- [123] K. Creamer and J. B. Lawrence, “Xist rna: a window into the broader role of rna in nuclear chromosome architecture,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 372, no. 1733, p. 20160360, 2017.

REFERENCES

- [124] M. Almeida, G. Pintacuda, O. Masui, Y. Koseki, M. Gdula, A. Cerase, D. Brown, A. Mould, C. Innocent, M. Nakayama *et al.*, “Pcgf3/5-prc1 initiates polycomb recruitment in x chromosome inactivation,” *Science*, vol. 356, no. 6342, pp. 1081–1084, 2017.
- [125] Y.-H. Xing, R.-W. Yao, Y. Zhang, C.-J. Guo, S. Jiang, G. Xu, R. Dong, L. Yang, and L.-L. Chen, “Slert regulates ddx21 rings associated with pol i transcription,” *Cell*, vol. 169, no. 4, pp. 664–678, 2017.
- [126] A. Postepska-Igielska, A. Giwojna, L. Gasri-Plotnitsky, N. Schmitt, A. Dold, D. Ginsberg, and I. Grummt, “Lncrna khps1 regulates expression of the proto-oncogene sphk1 via triplex-mediated changes in chromatin structure,” *Molecular cell*, vol. 60, no. 4, pp. 626–636, 2015.
- [127] J.-P. Couso and P. Patraquim, “Classification and function of small open reading frames,” *Nature reviews Molecular cell biology*, vol. 18, no. 9, pp. 575–589, 2017.
- [128] Y. Zhang, Z. Lin, and C. K. Kwoh, “Information theory-based feature selection: Minimum distribution similarity with removed redundancy,” in *International Conference on Computational Science*. Springer, 2020, pp. 3–17.
- [129] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.
- [130] J.-M. Claverie, O. Poirot, and F. Lopez, “The difficulty of identifying genes in anonymous vertebrate sequences,” *Computers & chemistry*, vol. 21, no. 4, pp. 203–214, 1997.
- [131] R. C. Deonier, S. Tavaré, and M. S. Waterman, *Computational genome analysis: an introduction*. Springer Science & Business Media, 2005.
- [132] J. W. Fickett, “Recognition of protein coding regions in dna sequences,” *Nucleic acids research*, vol. 10, no. 17, pp. 5303–5318, 1982.

REFERENCES

- [133] S. Nakagawa, Y. Niimura, T. Gojobori, H. Tanaka, and K.-i. Miura, “Diversity of preferred nucleotide sequences around the translation initiation codon in eukaryote genomes,” *Nucleic acids research*, vol. 36, no. 3, pp. 861–871, 2008.
- [134] A. V. Kochetov, “Aug codons at the beginning of protein coding sequences are frequent in eukaryotic mrnas with a suboptimal start codon context,” *Bioinformatics*, vol. 21, no. 7, pp. 837–840, 2005.
- [135] A. V. Pisarev, V. G. Kolupaeva, V. P. Pisareva, W. C. Merrick, C. U. Hellen, and T. V. Pestova, “Specific functional interactions of nucleotides at key-3 and+4 positions flanking the initiation codon with components of the mammalian 48s translation initiation complex,” *Genes & development*, vol. 20, no. 5, pp. 624–636, 2006.
- [136] K. A. Volkova OA, “Interrelations between the nucleotide context of human start aug codon, n-end amino acids of the encoded protein and initiation of translation.” *Journal of Biomolecular Structure and Dynamics*, vol. 27, no. 5, pp. 611–618, 2010.
- [137] K. Hanada, K. Akiyama, T. Sakurai, T. Toyoda, K. Shinozaki, and S.-H. Shiu, “sorf finder: a program package to identify small open reading frames with high coding potential,” *Bioinformatics*, vol. 26, no. 3, pp. 399–400, 2010.
- [138] E. M. Zdobnov and P. Bork, “Quantification of insect genome divergence,” *TRENDS in Genetics*, vol. 23, no. 1, pp. 16–20, 2007.
- [139] Y. Zhang, C. Jia, and C. K. Kwoh, “Predicting the interaction biomolecule types for lncrna: an ensemble deep learning approach,” *Briefings in Bioinformatics*, 2020.
- [140] H. Ling, K. Vincent, M. Pichler, R. Fodde, I. Berindan-Neagoe, F. J. Slack, and G. A. Calin, “Junk dna and the long non-coding rna twist in cancer genetics,” *Oncogene*, vol. 34, no. 39, pp. 5003–5011, 2015.
- [141] V. E. Villegas and P. G. Zaphiropoulos, “Neighboring gene regulation by antisense long non-coding rnas,” *International journal of molecular sciences*, vol. 16, no. 2, pp. 3251–3266, 2015.

REFERENCES

- [142] Y. Zhang, L. Zheng, B.-M. Xu, W.-H. Tang, Z.-D. Ye, C. Huang, X. Ma, J.-J. Zhao, F.-X. Guo, C.-M. Kang *et al.*, “Lnc rna-rp 11-714g18. 1 suppresses vascular cell migration via directly targeting lrp 2 bp,” *Immunology and cell biology*, vol. 96, no. 2, pp. 175–189, 2018.
- [143] Y. Wang, J. Hou, D. He, M. Sun, P. Zhang, Y. Yu, and Y. Chen, “The emerging function and mechanism of cernas in cancer,” *Trends in Genetics*, vol. 32, no. 4, pp. 211–224, 2016.
- [144] S. Zhou, L. Wang, Q. Yang, H. Liu, Q. Meng, L. Jiang, S. Wang, and W. Jiang, “Systematical analysis of lncrna–mrna competing endogenous rna network in breast cancer subtypes,” *Breast cancer research and treatment*, vol. 169, no. 2, pp. 267–275, 2018.
- [145] M. Cesana, D. Cacchiarelli, I. Legnini, T. Santini, O. Sthandier, M. Chinappi, A. Tramontano, and I. Bozzoni, “A long noncoding rna controls muscle differentiation by functioning as a competing endogenous rna,” *Cell*, vol. 147, no. 2, pp. 358–369, 2011.
- [146] M. A. Faghihi, M. Zhang, J. Huang, F. Modarresi, M. P. Van der Brug, M. A. Nalls, M. R. Cookson, G. St-Laurent, and C. Wahlestedt, “Evidence for natural antisense transcript-mediated inhibition of microRNA function,” *Genome biology*, vol. 11, no. 5, pp. 1–13, 2010.
- [147] M. R. Pickard and G. T. Williams, “Molecular and cellular mechanisms of action of tumour suppressor gas5 lncrna,” *Genes*, vol. 6, no. 3, pp. 484–499, 2015.
- [148] N. D. Damas, M. Marcatti, C. Côme, L. L. Christensen, M. M. Nielsen, R. Baumgartner, H. M. Gylling, G. Maglieri, C. F. Rundsten, S. E. Seemann *et al.*, “Snhg5 promotes colorectal cancer cell survival by counteracting stau1-mediated mrna destabilization,” *Nature communications*, vol. 7, no. 1, pp. 1–14, 2016.
- [149] M. Liang, K. Hu, C. He, J. Zhou, and Y. Liao, “Upregulated lncrna gm2044 inhibits male germ cell development by acting as mir-202 host gene,” *Animal cells and systems*, vol. 23, no. 2, pp. 128–134, 2019.

REFERENCES

- [150] M. Ballantyne, R. McDonald, and A. Baker, “Lncrna/microrna interactions in the vasculature,” *Clinical Pharmacology & Therapeutics*, vol. 99, no. 5, pp. 494–501, 2016.
- [151] T. Chiyomaru, S. Yamamura, S. Fukuhara, H. Yoshino, T. Kinoshita, S. Majid, S. Saini, I. Chang, Y. Tanaka, H. Enokida *et al.*, “Genistein inhibits prostate cancer cell growth by targeting mir-34a and oncogenic hotair,” *PloS one*, vol. 8, no. 8, p. e70372, 2013.
- [152] Q.-F. Yin, L. Yang, Y. Zhang, J.-F. Xiang, Y.-W. Wu, G. G. Carmichael, and L.-L. Chen, “Long noncoding rnas with snorna ends,” *Molecular cell*, vol. 48, no. 2, pp. 219–230, 2012.
- [153] D. Tian, S. Sun, and J. T. Lee, “The long noncoding rna, jpx, is a molecular switch for x chromosome inactivation,” *Cell*, vol. 143, no. 3, pp. 390–403, 2010.
- [154] Z. Lu, Q. C. Zhang, B. Lee, R. A. Flynn, M. A. Smith, J. T. Robinson, C. Davidovich, A. R. Gooding, K. J. Goodrich, J. S. Mattick *et al.*, “Rna duplex map in living cells reveals higher-order transcriptome structure,” *Cell*, vol. 165, no. 5, pp. 1267–1279, 2016.
- [155] Y. Yang, L. Wen, and H. Zhu, “Unveiling the hidden function of long non-coding rna by identifying its major partner-protein,” *Cell & bioscience*, vol. 5, no. 1, pp. 1–10, 2015.
- [156] L. Yang, Y. Tang, F. Xiong, Y. He, F. Wei, S. Zhang, C. Guo, B. Xiang, M. Zhou, N. Xie *et al.*, “Lncrnas regulate cancer metastasis via binding to functional proteins,” *Oncotarget*, vol. 9, no. 1, p. 1426, 2018.
- [157] X. Wang, S. Arai, X. Song, D. Reichart, K. Du, G. Pascual, P. Tempst, M. G. Rosenfeld, C. K. Glass, and R. Kurokawa, “Induced ncarnas allosterically modify rna-binding proteins in cis to inhibit transcription,” *Nature*, vol. 454, no. 7200, pp. 126–130, 2008.

REFERENCES

- [158] K. Daneshvar, M. B. Ardehali, I. A. Klein, F.-K. Hsieh, A. J. Kratkiewicz, A. Mahpour, S. O. Cancelliere, C. Zhou, B. M. Cook, W. Li *et al.*, “Lncrna digit and brd3 protein form phase-separated condensates to regulate endoderm differentiation,” *Nature cell biology*, vol. 22, no. 10, pp. 1211–1222, 2020.
- [159] L. Yang, C. Lin, W. Liu, J. Zhang, K. A. Ohgi, J. D. Grinstein, P. C. Dorrestein, and M. G. Rosenfeld, “ncrna-and pc2 methylation-dependent gene relocation between nuclear structures mediates gene activation programs,” *Cell*, vol. 147, no. 4, pp. 773–788, 2011.
- [160] T. Hung, Y. Wang, M. F. Lin, A. K. Koegel, Y. Kotake, G. D. Grant, H. M. Horlings, N. Shah, C. Umbricht, P. Wang *et al.*, “Extensive and coordinated transcription of noncoding rnas within cell-cycle promoters,” *Nature genetics*, vol. 43, no. 7, pp. 621–629, 2011.
- [161] X. Zhang, Y. Zhou, S. Chen, W. Li, W. Chen, and W. Gu, “Lncrna macc1-as1 sponges multiple mirnas and rna-binding protein ptbp1,” *Oncogenesis*, vol. 8, no. 12, pp. 1–13, 2019.
- [162] W. Yan, Z.-Y. Chen, J.-Q. Chen, and H.-M. Chen, “Lncrna neat1 promotes autophagy in mptp-induced parkinson’s disease through stabilizing pink1 protein,” *Biochemical and biophysical research communications*, vol. 496, no. 4, pp. 1019–1024, 2018.
- [163] D. D’Angelo, P. Mussnich, R. Sepe, M. Raia, L. Del Vecchio, P. Cappabianca, S. Pellecchia, S. Petrosino, S. Saggio, D. Solari *et al.*, “Rpsap52 lncrna is over-expressed in pituitary tumors and promotes cell proliferation by acting as mirna sponge for hmga proteins,” *Journal of molecular medicine*, vol. 97, no. 7, pp. 1019–1032, 2019.
- [164] W. H. Hudson and E. A. Ortlund, “The structure, function and evolution of proteins that bind dna and rna,” *Nature reviews Molecular cell biology*, vol. 15, no. 11, pp. 749–760, 2014.

REFERENCES

- [165] M. Han, Y. Gu, P. Lu, J. Li, H. Cao, X. Li, X. Qian, C. Yu, Y. Yang, X. Yang *et al.*, “Exosome-mediated lncrna afap1-as1 promotes trastuzumab resistance through binding with auf1 and activating erbb2 translation,” *Molecular cancer*, vol. 19, no. 1, pp. 1–18, 2020.
- [166] F. Ariel, L. Lucero, A. Christ, M. F. Mammarella, T. Jegu, A. Veluchamy, K. Mariappan, D. Latrasse, T. Blein, C. Liu *et al.*, “R-loop mediated trans action of the apolo long noncoding rna,” *Molecular cell*, vol. 77, no. 5, pp. 1055–1065, 2020.
- [167] S. Wang, H. Ke, H. Zhang, Y. Ma, L. Ao, L. Zou, Q. Yang, H. Zhu, J. Nie, C. Wu *et al.*, “Lncrna mir100hg promotes cell proliferation in triple-negative breast cancer through triplex formation with p27 loci,” *Cell death & disease*, vol. 9, no. 8, pp. 1–11, 2018.
- [168] L.-K. Wang, X.-F. Chen, D.-D. He, Y. Li, and J. Fu, “Dissection of functional lncrnas in alzheimer’s disease by construction and analysis of lncrna–mrna networks based on competitive endogenous rnas,” *Biochemical and biophysical research communications*, vol. 485, no. 3, pp. 569–576, 2017.
- [169] Q. Xiao, J. Luo, C. Liang, G. Li, J. Cai, P. Ding, and Y. Liu, “Identifying lncrna and mrna co-expression modules from matched expression data in ovarian cancer,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 2, pp. 623–634, 2018.
- [170] X. Chen, Y.-Z. Sun, N.-N. Guan, J. Qu, Z.-A. Huang, Z.-X. Zhu, and J.-Q. Li, “Computational models for lncrna function prediction and functional similarity calculation,” *Briefings in functional genomics*, vol. 18, no. 1, pp. 58–82, 2019.
- [171] X. Teng, X. Chen, H. Xue, Y. Tang, P. Zhang, Q. Kang, Y. Hao, R. Chen, Y. Zhao, and S. He, “Npinter v4. 0: an integrated database of ncRNA interactions,” *Nucleic acids research*, vol. 48, no. D1, pp. D160–D165, 2020.
- [172] C.-J. Liu, C. Gao, Z. Ma, R. Cong, Q. Zhang, and A.-Y. Guo, “lncrinter: a database of experimentally validated long non-coding rna interaction,” *Journal of genetics and genomics= Yi chuan xue bao*, vol. 44, no. 5, pp. 265–268, 2017.

REFERENCES

- [173] A. Frankish, M. Diekhans, A.-M. Ferreira, R. Johnson, I. Jungreis, J. Loveland, J. M. Mudge, C. Sisú, J. Wright, J. Armstrong *et al.*, “Gencode reference annotation for the human and mouse genomes,” *Nucleic acids research*, vol. 47, no. D1, pp. D766–D773, 2019.
- [174] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, “Cd-hit: accelerated for clustering the next-generation sequencing data,” *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012.
- [175] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K.-C. Chou, “Pse-in-one: a web server for generating various modes of pseudo components of dna, rna, and protein sequences,” *Nucleic acids research*, vol. 43, no. W1, pp. W65–W71, 2015.
- [176] Y. Lin, T. Liu, T. Cui, Z. Wang, Y. Zhang, P. Tan, Y. Huang, J. Yu, and D. Wang, “Rnainter in 2020: Rna interactome repository with increased coverage and annotation,” *Nucleic acids research*, vol. 48, no. D1, pp. D189–D197, 2020.
- [177] Q. Guo, L. Wang, L. Zhu, X. Lu, Y. Song, J. Sun, Z. Wu, J. Shi, Z. Wang, and X. Zhou, “The clinical significance and biological function of lncrna socar in serous ovarian carcinoma,” *Gene*, vol. 713, p. 143969, 2019.
- [178] S. Zhang, Y. Liang, Y. Wu, X. Chen, K. Wang, J. Li, X. Guan, G. Xiong, K. Yang, and Y. Bai, “Upregulation of a novel lncrna linc01980 promotes tumor growth of esophageal squamous cell carcinoma,” *Biochemical and biophysical research communications*, vol. 513, no. 1, pp. 73–80, 2019.
- [179] X. Liang, Z. Wu, S. Shen, Y. Niu, Y. Guo, J. Liang, and W. Guo, “Linc01980 facilitates esophageal squamous cell carcinoma progression via regulation of mir-190a-5p/myo5a pathway,” *Archives of biochemistry and biophysics*, vol. 686, p. 108371, 2020.
- [180] L.-L. Zhou, Y. Jiao, H.-M. Chen, L.-H. Kang, Q. Yang, J. Li, M. Guan, G. Zhu, F.-Q. Liu, S. Wang *et al.*, “Differentially expressed long noncoding rnas and regulatory mechanism of linc02407 in human gastric adenocarcinoma,” *World journal of gastroenterology*, vol. 25, no. 39, p. 5973, 2019.

REFERENCES

- [181] Y. Zhang, Y. Long, and C. K. Kwoh, “Deep learning based dna: Rna triplex forming potential prediction,” *BMC bioinformatics*, vol. 21, no. 1, pp. 1–13, 2020.
- [182] I. Antonov and Y. A. Medvedeva, “Purine-rich low complexity regions are potential rna binding hubs in the human genome,” *F1000Research*, vol. 7, 2018.
- [183] Y. Li, J. Syed, and H. Sugiyama, “Rna-dna triplex formation by long noncoding rnas,” *Cell chemical biology*, vol. 23, no. 11, pp. 1325–1333, 2016.
- [184] K.-M. Schmitz, C. Mayer, A. Postepska, and I. Grummt, “Interaction of noncoding rna with the rdna promoter mediates recruitment of dnmt3b and silencing of rrna genes,” *Genes & development*, vol. 24, no. 20, pp. 2264–2269, 2010.
- [185] P. Grote and B. G. Herrmann, “The long non-coding rna fendrr links epigenetic control mechanisms to gene regulatory networks in mammalian embryogenesis,” *RNA biology*, vol. 10, no. 10, pp. 1579–1585, 2013.
- [186] V. B. O’Leary, S. V. Ovsepian, L. G. Carrascosa, F. A. Buske, V. Radulovic, M. Niyazi, S. Moertl, M. Trau, M. J. Atkinson, and N. Anastasov, “Particle, a triplex-forming long ncRNA, regulates locus-specific methylation in response to low-dose irradiation,” *Cell reports*, vol. 11, no. 3, pp. 474–485, 2015.
- [187] M. Kalwa, S. Hänzelmann, S. Otto, C.-C. Kuo, J. Franzen, S. Joussen, E. Fernandez-Rebollo, B. Rath, C. Koch, A. Hofmann *et al.*, “The lncRNA ho-tair impacts on mesenchymal stem cells via triple helix formation,” *Nucleic acids research*, vol. 44, no. 22, pp. 10 631–10 643, 2016.
- [188] Z. Zhao, N. Sentürk, C. Song, and I. Grummt, “lncRNA papas tethered to the rdna enhancer recruits hypophosphorylated chd4/nurd to repress rrna synthesis at elevated temperatures,” *Genes & development*, vol. 32, no. 11-12, pp. 836–848, 2018.
- [189] N. Sentürk Cetin, C.-C. Kuo, T. Ribarska, R. Li, I. G. Costa, and I. Grummt, “Isolation and genome-wide characterization of cellular dna: Rna triplex structures,” *Nucleic acids research*, vol. 47, no. 5, pp. 2306–2321, 2019.

REFERENCES

- [190] C. Navarro, C. Cano, M. Cuadros, A. Herrera-Merchan, M. Molina, and A. Blanco, “A mechanistic study of lncrna fendrr regulation of foxf1 lung cancer tumor suppressor,” in *International conference on bioinformatics and biomedical engineering*. Springer, 2016, pp. 781–789.
- [191] A. A. Ageeli, K. R. McGovern-Gooch, M. M. Kaminska, and N. J. Baird, “Finely tuned conformational dynamics regulate the protective function of the lncrna malat1 triple helix,” *Nucleic acids research*, vol. 47, no. 3, pp. 1468–1481, 2019.
- [192] K. Mishra and C. Kanduri, “Understanding long noncoding rna and chromatin interactions: what we know so far,” *Non-coding RNA*, vol. 5, no. 4, p. 54, 2019.
- [193] K. Inamura, “Major tumor suppressor and oncogenic non-coding rnas: clinical relevance in lung cancer,” *Cells*, vol. 6, no. 2, p. 12, 2017.
- [194] D. Liu, R. Mewalal, R. Hu, G. A. Tuskan, and X. Yang, “New technologies accelerate the exploration of non-coding rnas in horticultural plants,” *Horticulture research*, vol. 4, no. 1, pp. 1–8, 2017.
- [195] F. Russo, F. Scoyni, A. Fatica, M. Pellegrini, A. Ferro, A. Pulvirenti, and R. Giugno, “Circulating noncoding rnas as clinical biomarkers,” in *Epigenetic biomarkers and diagnostics*. Elsevier, 2016, pp. 239–258.
- [196] D. P. Clark, N. J. Pazdernik, and M. R. McGehee, “Chapter 19 - noncoding rna,” in *Molecular Biology (Third Edition)*, third edition ed., D. P. Clark, N. J. Pazdernik, and M. R. McGehee, Eds. Academic Cell, 2019, pp. 604–621. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128132883000197>
- [197] M. Chaabane, R. M. Williams, A. T. Stephens, and J. W. Park, “circdeep: deep learning approach for circular rna classification from other long non-coding rna,” *Bioinformatics*, vol. 36, no. 1, pp. 73–80, 2020.
- [198] M. Niu, J. Zhang, Y. Li, C. Wang, Z. Liu, H. Ding, Q. Zou, and Q. Ma, “Cirrnapl: a web server for the identification of circrna based on extreme learning machine,” *Computational and structural biotechnology journal*, vol. 18, pp. 834–842, 2020.

REFERENCES

- [199] G. Zhang, Y. Deng, Q. Liu, B. Ye, Z. Dai, Y. Chen, and X. Dai, “Identifying circular rna and predicting its regulatory interactions by machine learning,” *Frontiers in genetics*, vol. 11, p. 655, 2020.
- [200] S. Ye, L. Yang, X. Zhao, W. Song, W. Wang, and S. Zheng, “Bioinformatics method to predict two regulation mechanism: Tf–mirna–mrna and lncrna–mirna–mrna in pancreatic cancer,” *Cell biochemistry and biophysics*, vol. 70, no. 3, pp. 1849–1858, 2014.