A DEEP LEARNING MODEL FOR PREDICTING LONG NON-CODING RNA AND MESSENGER RNA WITH MODEL INTERPRETATION

Mr. Rattaphon Lin

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

Copyright of Chulalongkorn University

แบบจำลองการเรียนรู้เชิงลึกสำหรับการทำนายอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัสและอาร์เอ็นเอนำรหัสพร้อมการตีความแบบจำลอง

นายรัฐพล หลิน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	A DEEP LEARNING MODEL FOR PREDICTING LONG NON-CODING RNA AND MESSENGER RNA WITH MODEL INTERPRETATION
By	Mr. Rattaphon Lin
Field of Study	Computer Science
Thesis Advisor	Associate Professor DUANGDAO WICHADAKUL

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

-- Dean of the FACULTY OF ENGINEERING
(Professor SUPOT TEACHAVORASINSKUN)

THESIS COMMITTEE

-- Chairman
(Associate Professor KRERK PIROMSOPA)

-- Thesis Advisor
(Associate Professor DUANGDAO WICHADAKUL)

-- Examiner
(Associate Professor PEERAPON VATEEKUL)

-- External Examiner
(Associate Professor Benchaphon Limthanmaphon)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

รัฐพล หลิน : แบบจำลองการเรียนรู้เชิงลึกสำหรับการทำนายอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัสและอาร์เอ็นเอนำรหัสพร้อมการตีความแบบจำลอง. (A DEEP LEARNING MODEL FOR PREDICTING LONG NON-CODING RNA AND MESSENGER RNA WITH MODEL INTERPRETATION) อ.ที่ปรึกษาหลัก : รศ. ดร.ดวงดาว วิชาดากุล

อาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัส (long non-coding RNA: lncRNA) มีบทบาทสำคัญในกระบวนการทางชีววิทยาและยังพบว่ามีส่วนเกี่ยวข้องกับการเกิดโรคต่าง ๆ จากการพัฒนาเทคโนโลยีการค้นหาลำดับเบสในปัจจุบันนำไปสู่การค้นพบสายอาร์เอ็นเอที่ยังไม่ถูกระบุประเภทเป็นจำนวนมาก การจำแนกสายอาร์เอ็นเอเหล่านี้ด้วยการทดลองทางชีววิทยามีค่าใช้จ่ายสูงและใช้เวลานาน ดังนั้นการจำแนกด้วยเครื่องมือหรือซอฟต์แวร์จากการคำนวณจึงเป็นอีกทางเลือกที่ประหยัดและรวดเร็ว เครื่องมือทางคอมพิวเตอร์ที่ใช้ในการจำแนกอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัสมีจำนวนมากถึงอย่างไรก็ตามเครื่องมือเหล่านี้ไม่มีการอธิบายคุณลักษณะสำคัญที่ใช้ในการจำแนก ดังนั้นงานวิจัยนี้นำเสนอเครื่องมือใหม่ชื่อ Xlnc1DCNN ที่สามารถจำแนกอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัสและอาร์เอ็นเอนำรหัสพร้อมทั้งอธิบายผลลัพธ์จากการทำนาย แบบจำลองนี้ถูกพัฒนาด้วยแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันหนึ่งมิติ และใช้ DeepSHAP ในการอธิบายคุณลักษณะที่ใช้ในการจำแนก จากผลการประเมินแบบจำลองด้วยข้อมูลสายอาร์เอ็นเอมนุษย์ชุดทดสอบพบว่า Xlnc1DCNN มีประสิทธิภาพที่เหนือกว่าเครื่องมืออื่น ๆ ด้วยตัวชี้วัดความแม่นยำ (accuracy) และ F1-score อีกทั้งยังมีความเป็นนัยทั่วไป (generalization) สำหรับสปีชีส์อื่น ๆ ผลลัพธ์จากการอธิบายแบบจำลองพบว่าการจำแนกอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัส พบรูปแบบที่ไม่อนุรักษ์ พบรูปแบบนิวคลีโอไทด์สายสั้นที่ไม่ทราบฟังก์ชัน หรือเป็นบริเวณที่พบเฉพาะทรานส์เมมเบรนฮีลิกซ์ ในขณะที่ส่วนอาร์เอ็นเอนำรหัสบริเวณที่สำคัญจะพบโปรตีนโดเมนหรือวงศ์โปรตีน อีกทั้งผลลัพธ์ที่แบบจำลองทายผิดพบบรรณนิทัศน์ที่ขัดแย้งระหว่างฐานข้อมูลสาธารณะ โดยพบโปรตีนโดเมนหรือวงศ์โปรตีนในอาร์เอ็นเอสายยาวที่ไม่ถูกแปลรหัสหรือพบบริเวณที่มีความผิดปกติของสายโปรตีน เครื่องมือนี้เปิดให้ใช้งานแบบสาธารณะที่ https://github.com/cucpbioinfo/Xlnc1DCNN

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต ....
ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6170952721 : MAJOR COMPUTER SCIENCE

KEYWORD: SHAP (SHapley additive exPlanations), Deep learning, explainable artificial intelligence (XAI), Long non-coding RNA, one-dimensional convolutional neural network (1D CNN)

Rattaphon Lin : A DEEP LEARNING MODEL FOR PREDICTING LONG NON-CODING RNA AND MESSENGER RNA WITH MODEL INTERPRETATION. Advisor: Assoc. Prof. DUANGDAO WICHADAKUL

Long non-coding RNAs (lncRNAs) play important roles in many biological processes and are found to be associated with several diseases. The development of next-generation sequencing technologies has discovered numerous unannotated transcripts. However, classifying these unannotated transcripts by using biological experiments is very time-consuming and expensive. Thus, a computational approach is considered as an alternative solution which is faster and cheaper. Many existing lncRNA identification tools are available, these tools lack an explanation of which features contributed to their prediction results. Here, we present Xlnc1DCNN, a tool for distinguishing long non-coding RNAs (lncRNAs) from protein-coding transcripts (PCTs) together with a prediction explanation. We developed the model by using a one-dimensional convolutional neural network integrated with DeepSHAP. On the human test dataset, we showed that Xlnc1DCNN outperformed other lncRNA identification tools in terms of accuracy and F1-score and had a generalization to other species. We also explained the prediction result to understand further how the model makes predictions. The explanation results revealed that most of the lncRNA transcripts were identified without any conserved regions, short patterns with unknown functions, or only regions of transmembrane helices while protein-coding transcripts were mostly identified with protein domains or families. Some of the incorrect predictions of the model also found inconsistent annotations among the public databases with lncRNA transcripts containing protein domains, protein families, or intrinsically disordered regions (IDRs). Xlnc1DCNN is freely available at https://github.com/cucpbioinfo/Xlnc1DCNN.

Field of Study: Computer Science Student's Signature ....

Academic Year: 2021 Advisor's Signature ....

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ACKNOWLEDGEMENTS

First of all, I could not have undertaken this dissertation without my advisor, Assoc. Prof. Duangdao Wichadakul, who always keeps supporting me during my hard times, guiding me in the right direction without being lost and patiently teaching me how to conduct research. Writing this dissertation made me realize that there is no gain without pain, and I had to endure the storm before I could enjoy the rainbow. I also would like to express my deepest gratitude to my dissertation committees: Assoc. Prof. Krerk Piromsopa, Assoc. Prof. Peerapon Vateekul, and Assoc. Prof. Benchaphon Limthanmaphon.

This work also could not be finished without the support of all staff in the computer engineering department and I greatly appreciate the financial support from the Ratchadaphiseksomphot Endowment Fund Part of the “Research Grant for New Scholar CU Researcher’s Project”, grant number (RGN_2559_025_06_21).

Lastly, I would like to thank my family for supporting me, especially my mother, who encouraged me to continue fighting, and everyone who supported me and made me believe in myself.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Rattaphon Lin

TABLE OF CONTENTS

Page

ABSTRACT (THAI) iii

ABSTRACT (ENGLISH) iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

Chapter 1 1

Introduction 1

Chapter 2 4

Literature Review 4

2.1 Protein-coding and long non-coding RNA identification tools 4

2.1.1 CPAT 5

2.1.2 PLEK 5

2.1.3 FEELNC 5

2.1.4 CPC2 6

2.1.5 LncADeep 6

2.1.6 CNIT 6

2.1.7 RNAsamba 7

2.1.8 lncRNA_Mdeep 7

2.1.9 Summary 7

2.2 Application of 1D-CNN 7

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2.1 Electrocardiogram (ECG) Monitoring........8
2.2.2 Summary........10
2.3 Machine Learning Model Interpretation Methods........10
2.3.1 Additive feature attribution methods........11
2.3.2 LIME........11
2.3.3 DeepLIFT........12
2.3.4 SHAP........13
2.3.5 Kernel SHAP (LIME + Shapley values)........14
2.3.6 DeepSHAP (DeepLIFT + Shapley values)........15
2.3.7 Summary........15
Chapter 3........16
Methods........16
3.1 Data Preparation and Preprocessing........17
3.2 Model Architecture and Training Procedure........18
3.2.1 Model Hyperparameters Optimization........19
3.2.2 Model Hyperparameters in Details........20
3.2.3 Details of learning........21
3.3 Model Selection........21
3.4 Model Interpretation method........23
3.5 Evaluation........25
3.5.1 Model Evaluation Metrics........25
3.5.2 Interpretation Evaluation Method........26
Chapter 4........27
Results........27

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.1 Model Evaluation Results....27

4.1.1 Performance Evaluation on Human Test set....27

4.1.2 Performance Evaluation on Cross-Species Datasets....29

4.2 Model Interpretation Results....30

4.2.1 True Positive Sequences....31

4.2.2 True Negative Sequences....35

4.2.3 False Positive Sequences....36

4.2.4 False Negative Sequences....38

4.3 Sequences Annotation results....39

Chapter 5....42

Discussion....42

Conclusion....43

Future Work....44

REFERENCES....45

Appendix....50

VITA....53

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF TABLES

Page

Table 1: The overview of selected tools to be compared with the proposed model..4

Table 2: The 2nd proposed 1D-CNN architecture from [22]..8

Table 3: The parameters of 1D-CNN architecture for net B from [20] 9

Table 4: The summary of human datasets from GENCODE and LNCipedia 18

Table 5: Hyperparameters of the proposed 1D-CNN architecture 19

Table 6: Model performance of all candidate models...22

Table 7: Software version 25

Table 8: Evaluation results of all tools on the human test set..28

Table 9: Accuracy of all models on cross-species datasets...29

Table 10: Summary of test set sequences annotated with InterPro entries..................40

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF FIGURES

Page

Figure 1: The architecture of proposed 1D-CNN for net B from [20] 9

Figure 2: The proposed 1D-CNNs structure from [21] 10

Figure 3: The red cross, an instance x, is an instance of interest to be explained. The black curve represents the decision boundary of the complex black-box model f. LIME generates new training data (blue and red dots) then weights them according to the distance between an instance x (shown by size). LIME then trains a simple surrogate model g with new training data. The dashed line mask the decision boundary of the explanation model g 12

Figure 4: An example of a long non-coding RNA transcript from GENCODE in FASTA format 16

Figure 5: An example of a protein-coding transcript from GENCODE in FASTA format 17

Figure 6: (A) Distribution of target class (B) Distribution of the sequences' length 18

Figure 7: The model performance on the validation set. (A) The model performance for each kernel size (B) The model accuracy of a kernel size of 57 for each stride size 20

Figure 8: Transformation methods for 2D-CNN (A) Basic (B) Spiral (C) Pairwise Features 22

Figure 9: The explanation result from each size of the background sample 23

Figure 10: The process to obtain SHAP values for each amino acid 24

Figure 11: (A) ROC curve of all tools and their AUC on the human test set (B) Accuracy of all tools for any range of sequence lengths on the human test set 28

Figure 12: Receiver operating characteristic curves and AUCs of nine models on the datasets of (A) mouse (B) gorilla (C) cow and (D) chicken 30

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure13: Explanation results of Xlnc1DCNN highlighted weak signals on TP sequences (A) ENST00000658844.1, a lncRNA sequence obtained from GENCODE and (B) lnc-REXO4-2:1, a lncRNA sequence obtained from LNCipedia. 32

Figure 14: (A) Explanation result of Xlnc1DCNN and (B) the prediction result of the transmembrane helices by TMHMM program on the true positive sequence, lnc-NXNL1-3:11 which corresponds to the explanation result of Xlnc1DCNN. 33

Figure 15: (A) Explanation result of Xlnc1DCNN and (B) the signal peptide identified by InterPro on the true positive sequence, ENST00000663782.1 obtained from GENCODE. 34

Figure 16: The explanation results of Xlnc1DCNN on true positive sequences of the (A) lnc-THTPA-2:32, (B) lnc-VIM-3:1 obtained from LNCipedia. 35

Figure 17: Comparison between the explanation results of Xlnc1DCNN on TN sequences (A) ENST00000528724.5, (B) ENST00000593088.5, and (C) ENST00000589852.5 protein-coding transcripts and (D) the prediction result of TMHMM program on the ENST00000528724.5 (E) the KRAB (Krüppel associated box) domain identified by Pfam within the ENST00000593088.5 and (F) the FAM32A family identified by InterPro within the ENST00000589852.5 transcripts. 37

Figure 18: Comparison between (A) the explanation result of Xlnc1DCNN on the ENST00000408930.6 protein-coding transcript, predicted as a lncRNA (B) the identification result from Pfam and (C) the identification result from InterPro. 38

Figure 19: Comparison between the explanation result of Xlnc1DCNN on the long non-coding RNA transcripts (A) lnc-SIGIRR-2:1 and (B) ENST00000616537.4, predicted as mRNAs (C) the Anoctamin family within the lnc-SIGIRR-2:1 transcript and (D) the Taxilin family within the ENST00000616537.4 transcript identified by InterPro. 39

Figure 20: Mean of absolute SHAP values for (A) single nucleotide (B) dinucleotide and (C) trinucleotide, indicating the impact of each genetic code on the model prediction as lncRNA or mRNA. 43

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter 1

Introduction

Long non-coding RNA sequences (lncRNAs) are RNAs longer than 200 nucleotides that are not translated into proteins. lncRNAs have many crucial roles in several biological processes, such as gene silencing, gene regulation, gene expression, acting as molecular scaffolds, chromatin remodeling, etc. [1-3], and some studies also found that lncRNAs have been linked to human diseases such as diabetes or cancers [4-7]. A large volume of unannotated transcripts has been discovered by the advancement of next-generation sequencing technology, i.e., RNA Sequencing (RNA-Seq) [8, 9]. Classifying unannotated transcripts through biological experimentation is very time-consuming and cost intensive. Therefore, numerous studies have proposed computational approaches to identify these unclassified sequences that are quicker and cheaper.

The existing computational approaches, e.g., CPC2 [10], CNIT [11], PLEK [12], CPAT [13], FEELnc [14], RNAsamba [15], LncADeep [16], and lncRNA_Mdeep [17], obtained training features by using feature extraction methods. Most of them trained their proposed algorithm with similar features, such as ORF length, Fickett and hexamer scores, and then combined with additional extracted structural and sequence features. However, none of them described how each feature affected the model's predicted outcomes.

Deep learning algorithms have gained a lot of popularity, especially for datasets with many data points and dimensions, since the algorithms will automatically learn the complex patterns within features as they are trained. Convolutional neural networks (CNNs), especially the 2D-CNNs, have dominated many computer vision applications, e.g., image classification, image segmentation, and image detection [18], because of their excellent capability for automatically learning and extracting complex patterns within the input data. While 1D-CNN also achieved state-of-the-art

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

performance in several applications, such as ECG monitoring and speech recognition which outperformed the traditional approaches [19]. For detecting irregular heartbeats applications, [20-22] demonstrated that using only a simple 1D-CNN architecture could achieve excellent prediction accuracy without explicitly addressing and extracting features for training their models.

Understanding the decision-making of any artificial intelligent model can help users trust and comprehend the model's prediction. It can assist in illustrating what the models perceive and explain how these perceptions relate to fundamental human knowledge. In general, most complicated black-box models (e.g., deep neural networks, ensemble models, gradient boosting tree algorithms) have demonstrated superior learning performance, but most are uninterpretable. Explainable artificial intelligence (XAI) has lately become a significant research area [23] that aims to understand these complicated black-box models. LIME [24] and SHAP [25] is one of the approaches to obtain an explanation from a complex black-box model. LIME explained each individual prediction by creating a local surrogate model while SHAP (Shapley Additive exPlanations) introduced SHAP values representing the model's unified measure of feature importance and SHAP value estimation methods. Another method to explain any deep learning models is DeepSHAP. DeepSHAP [26] was developed based on the relationship between DeepLIFT [27] and the original SHAP to explain the deep learning model and further refined and extended with stacks of mixed model types and relative background distributions.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Classifying lncRNA and mRNA sequences based on training features still have some ambiguities, given the promising outcomes of 1D-CNN in prior applications. In this thesis, we present Xlnc1DCNN, a long non-coding RNA identification tool developed by 1D-CNN integrated with a prediction explanation. Xlnc1DCNN was solely trained by nucleotide sequences without applying any feature extraction method. We showed that Xlnc1DCNN outperformed other existing tools in terms of accuracy and F1-Score on the human dataset while maintaining generalization among other testing species. We also provide the insight obtained from the explanation results. DeepSHAP was used to generate SHAP values that represented what our model learned. Using our in-house Python code, we then visualized each nucleotide's contribution and amino acid's contribution. The explanation of true positives (i.e., lncRNA transcript sequences) revealed that the model classified a sequence as lncRNA if it comprised only transmembrane helices or an N-terminal signal peptide and no important regions. The explanation of true negatives (i.e., mRNA transcript sequences) revealed that the model could learn and utilize protein domains or families within the input sequences to predict the sequences as mRNAs. The explanation of false positives (i.e., mRNA transcript sequences predicted as lncRNA transcript sequences) revealed that the model was incapable of capturing important regions representing protein domains or families or identifying important regions contributing to both lncRNA and mRNA. A small number of false-positive sequences were also found with varying transcript types across databases. Finally, the explanation for false negatives (i.e., lncRNA transcript sequences predicted as mRNA transcript sequences) revealed that the model captured protein domains or families within these lncRNA sequences, hence, misclassifying them as mRNAs.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Chapter 2

Literature Review

In this chapter, we will discuss about previous works that are related to this research. First, we will go through the existing protein-coding and long non-coding RNA identification tools. Second, we will explore some previous work about the application of 1D-CNN. The last topic will be about methods for the interpretation of machine learning and deep learning models.

2.1 Protein-coding and long non-coding RNA identification tools

In this section, we will discuss about the existing tools for protein-coding and long non-coding RNAs identification. We selected eight tools: CPAT, PLEK, FEELNC, CPC2, LncADeep, CNIT, RNAsamba, and lncRNA_Mdeep to compare with our proposed method. All tools were chosen by their performance, availability, and reliability. The overview of each tool is shown in **Table 1**.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 1: The overview of selected tools to be compared with the proposed model

Software	Algorithm	Supported Species	Published Year	Web Interface
CPAT	Logistic Regression	Human, Mouse, Fly, Zebrafish	2013	Yes
PLEK	SVM	All	2014	No
FEELNC	Random Forest	All	2017	No
CPC2	SVM	Human, Chimpanzee, Mouse, Zebrafish, Xenopus, Fruit fly	2017	Yes
LncADeep	Deep Learning	All	2018	No

Software	Algorithm	Supported Species	Published Year	Web Interface
CNIT	XGBoost	Animal, Plant	2019	Yes
RNAsamba	Deep Learning	All	2020	Yes
lncRNA_Mdeep	Deep Learning	All	2020	No

To comprehend the method behind each tool, we will briefly review these eight tools from the aspects of their machine learning algorithms and the training features.

2.1.1 CPAT

CPAT [13] is an alignment-free coding potential assessment tool based on a logistic regression model. CPAT is trained with four basic sequence features that are calculated from the sequence directly, including size of open reading frame (ORF), ORF coverage, Fickett TESTCODE score, and hexamer score. CPAT also supports a web interface for users to submit sequences and receive back the sequence prediction results.

2.1.2 PLEK

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

PLEK [12] is another alignment-free computational tool for protein-coding and non-coding RNA identification. PLEK uses a *k*-mer scheme and sliding windows to create features from RNA sequences. PLEK then was trained with an SVM model using a radial basis function kernel.

2.1.3 FEELNC

FEELNC [14] is also an alignment-free tool based on a Random Forest model that was trained with the general sequence features (multi *k*-mer frequencies between protein-coding and lncRNAs, relaxed ORF, and RNA sequence length). FEELNC also can predict and analyze coding potential even without lncRNAs learning dataset.

2.1.4 CPC2

CPC2 [10] is the upgraded version of CPC1 [28], which runs ~1000 times faster and more accurate. CPC2 used four intrinsic features, including Fickett score, ORF length, ORF integrity, and isoelectric point. CPC2 then employed LIBSVM package to train an SVM model with the standard radial basis function kernel (RBF kernel). The training data of CPC2 were human protein-coding sequences and non-coding sequences from RefSeq. CPC2 also has a web interface for coding potential calculator that allows users to submit sequences.

2.1.5 LncADeep

LncAdeep [16] proposed a tool for lncRNA identification and functional annotation. For lncRNA identification, LncAdeep trained with sequence intrinsic features and homology features. LncAdeep was implemented by using a deep belief network (DBN) and built as a stack of restricted Boltzmann machines (RBMs).

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.6 CNIT

CNIT [11] is the upgraded version from CNCI, which runs ~200 times faster and has better accuracy in more species, especially for plants. CNIT used the nucleotide sequences intrinsic composition, a nucleotide triplet called ANT, as training features. A total of 67 features were used to train the XGBoost models. CNIT also provides a web server interface and a stand-alone package.

2.1.7 RNAsamba

RNAsamba [15] is a tool for predicting the coding potential based on a novel neural network classification model. RNAsamba used the IGLOO architecture and then was solely trained with sequence information. RNAsamba also provides a docker image, a web server interface, and a stand-alone package.

2.1.8 lncRNA_Mdeep

lncRNA_Mdeep [17] proposed an alignment-free multimodal deep learning framework for lncRNA identification. lncRNA_Mdeep used three main features including OFH features (the length and coverage of ORF, Fickett score, and Hexamer score), k-mer features, and one-hot encoding with a simple CNN layer then concatenated these features and trained using deep neural networks.

2.1.9 Summary

All the reviewed tools have relied on feature extraction methods. Most used similar features, such as the Fickett score or ORF length. None of them explain how each feature contributed to the prediction output. For example, the ORF length and Fickett score range that the sequences will classify as lncRNAs or mRNAs. Hence, the explanation of each feature that contributes to the prediction output remains challenging.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 Application of 1D-CNN

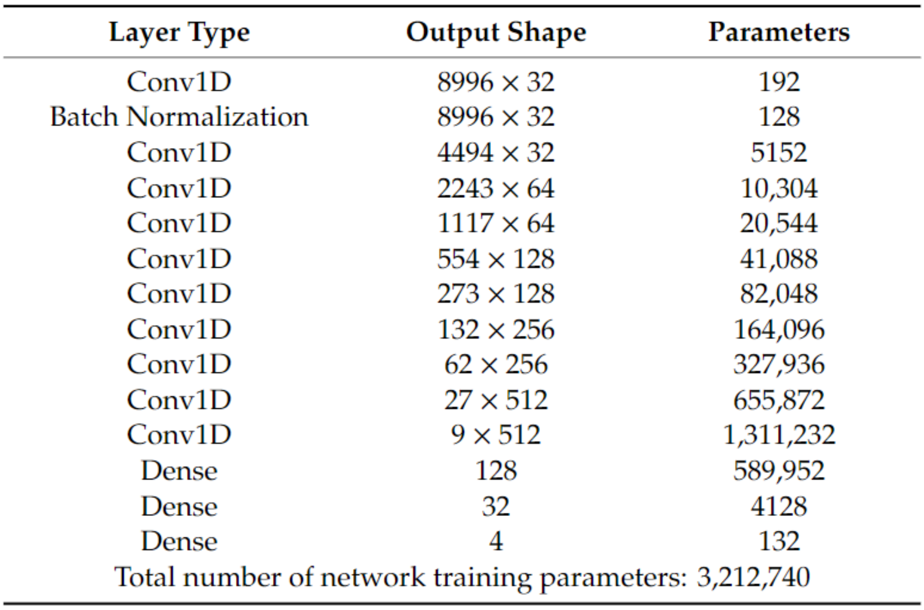
The CNNs are widely used due to their good capability for extracting features from input data. In this section, we will present the applications of 1D-CNN.

2.2.1 Electrocardiogram (ECG) Monitoring

Cardiovascular disease (CVD) is a leading cause of death worldwide. One of the indicator symptoms of cardiovascular disease is irregular heartbeats. Electrocardiogram (ECG) is a common approach to diagnose many heart problems by using an electrical signal to monitor and analyze the heart’s health.

Several studies for detecting irregular heartbeats have been proposed. C.-H. Hsieh *et al*. [22] illustrated that a simple 1D-CNN architecture (**Table 2**) can outperform the existing DL-based methods for detecting an arrhythmia from time-series ECG signal.

Table 2: The 2nd proposed 1D-CNN architecture from [22]



Layer Type	Output Shape	Parameters
Conv1D	8996 × 32	192
Batch Normalization	8996 × 32	128
Conv1D	4494 × 32	5152
Conv1D	2243 × 64	10,304
Conv1D	1117 × 64	20,544
Conv1D	554 × 128	41,088
Conv1D	273 × 128	82,048
Conv1D	132 × 256	164,096
Conv1D	62 × 256	327,936
Conv1D	27 × 512	655,872
Conv1D	9 × 512	1,311,232
Dense	128	589,952
Dense	32	4128
Dense	4	132
Total number of network training parameters: 3,212,740		

U. R. Acharya *et al.* [20] proposed two CNN architectures (net A and net B). A 1D-CNN architecture and parameters are shown in **Figure 1** and **Table 3**. The net B CNN architectures from [20] yielded 94.90% accuracy, 99.13% sensitivity, and 81.44 % specificity for five seconds of ECG duration.

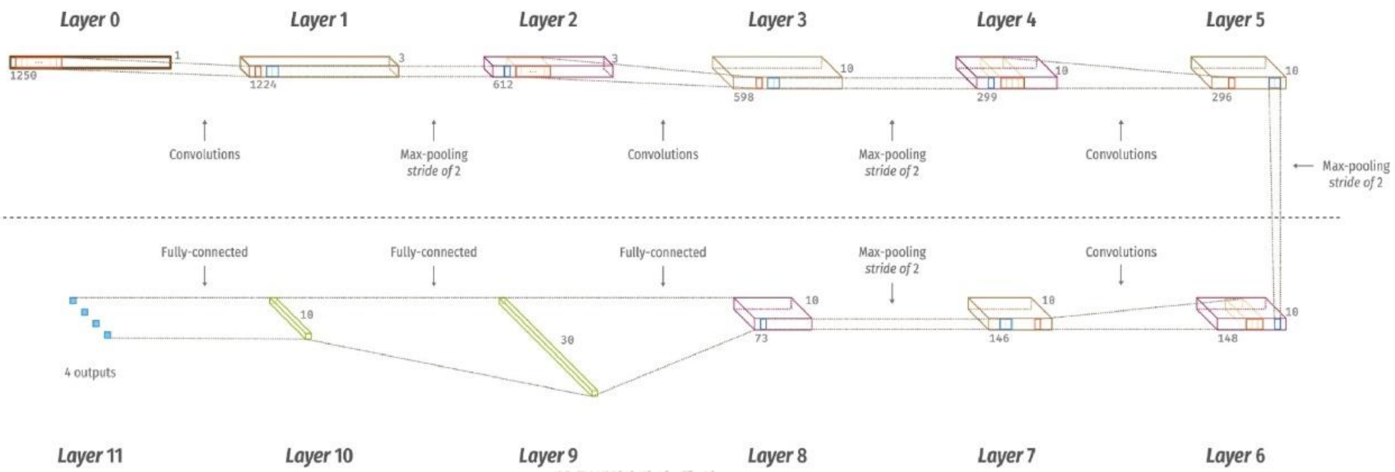


Figure 1: The architecture of proposed 1D-CNN for net B from [20]

Table 3: The parameters of 1D-CNN architecture for net B from [20]

The details of CNN structure for net B.

Layers	Type	Number of neurons (output layer)	Kernel size for each output feature map	Stride
0–1	Convolution	1224 × 3	27	1
1–2	Max-pooling	612 × 3	2	2
2–3	Convolution	598 × 10	15	1
3–4	Max-pooling	299 × 10	2	2
4–5	Convolution	296 × 10	4	1
5–6	Max-pooling	148 × 10	2	2
6–7	Convolution	146 × 10	3	1
7–8	Max-pooling	73 × 10	2	2
8–9	Fully-connected	30	–	–
9–10	Fully-connected	10	–	–
10–11	Fully-connected	4	–	–

Another method to detect an abnormal heartbeat, Fenli *et al*. [21], used high-frequency murmurs from phonocardiogram (PCG) as a training set to classify healthy and confirmed cardiac patients before the ECG signals from patients will become irregular. A simple 1D-CNN architecture was also used in [21] as a learning model for detecting irregular heartbeat, as shown in **Figure 2**.

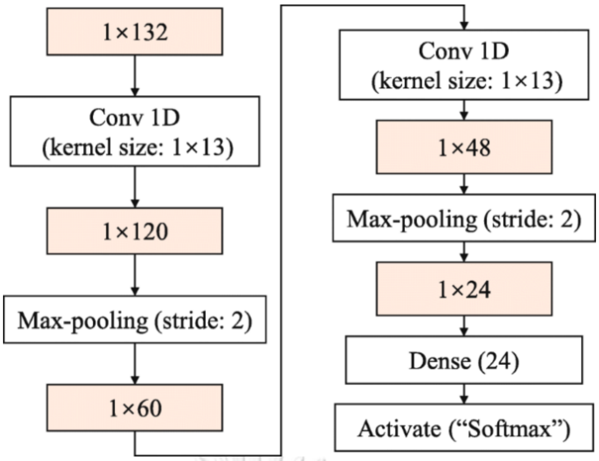


Figure 2: The proposed 1D-CNNs structure from [21]

2.2.2 Summary

The applications for detecting irregular heartbeats [20-22] showed that they could achieve high accuracy using a simple 1D-CNN without applying any further feature extraction methods from their datasets.

2.3 Machine Learning Model Interpretation Methods

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

In this final topic, we will describe some works about the machine learning interpretation methods that would be beneficial for interpreting our proposed model.

Explaining simple machine learning models, such as decision trees or linear models, are easy to understand and can be interpreted by their decision nodes or weights. On the other hand, complex black-box models such as deep neural networks or ensemble models cannot be used for interpretation. Hence, it is challenging to understand the reason behind each prediction result. In order to understand the underlying cause for each prediction from complex black-box models, many interpretation methods have been proposed as a simplified explanation model for interpreting the original complex model.

Recently, several interpretation methods have been published that correspond to the definition of additive feature attribution methods [25], such as LIME [24], DeepLIFT [27], SHAP, and DeepSHAP [25, 26].

2.3.1 Additive feature attribution methods

To explain a model, an additive feature attribution approach combines the impacts of all input feature attribution [25]. The explanation model g can be explained as a linear function of the binary variable. We can approximate the output of z' with the attribution value for each ϕ_i from the model g using the following Equation (1).

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

where $\phi_i \in \mathbb{R}$ is the feature attribution for a feature i, $z' \in \{0,1\}^M$, and M is the number of simplified input features. If the feature i is present then $z'_i = 1$, otherwise $z'_i = 0$.

2.3.2 LIME

Local Interpretable Model-Agnostic Explanations (LIME) [24] is a method to explain any black-box machine learning model. The authors proposed a local surrogate model that is trained to explain individual prediction of the underlying black-box model.

Figure 3 demonstrates how LIME works, suppose we want to explain the prediction output of instance x, where $x \in \mathbb{R}$, from the complex black-box model f. LIME works by training a new local surrogate model g, where g can be a linear model or decision tree. Using newly generated training data x', where x' are permuted samples around instance x from a distance function π_x, the model g should yield a good prediction result for local approximation but not necessary in global approximation. This kind of approximation is called local fidelity.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

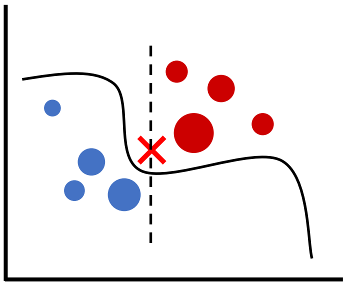


Figure 3: The red cross, an instance x, is an instance of interest to be explained. The black curve represents the decision boundary of the complex black-box model f. LIME generates new training data (blue and red dots) then weights them according to the distance between an instance x (shown by size). LIME then trains a simple surrogate model g with new training data. The dashed line mask the decision boundary of the explanation model g

LIME produces the explanation ξ from an instance x by minimizing the loss function $\mathcal{L}$ (e.g., weighted square loss, mean squared error). The loss function measures how the prediction from the surrogate model g is close to the prediction of the original model f and then penalizes with the model complexity $\Omega(g)$. The explanation $\xi(x)$ can be expressed by the following Equation (2).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2}$$

2.3.3 DeepLIFT

DeepLIFT (Deep Learning Important FeaTures) [27] is a method to explain the deep learning model by differencing the input data from a selected ‘reference’, where a ‘reference’ is selected according to the current objective for solving the problem. DeepLIFT uses the summation-to-delta property as shown in Equation (3) to obtain the explanation scores,

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t \tag{3}$$

where $\Delta t = t - t^0$, t is the prediction output from the model, t^0 is the reference, x_i is neuron i to n that are neceesary to compute t, and $C_{\Delta x_i \Delta t}$ is the amount of difference between the reference t^0 from the output t.

If we replace $C_{\Delta x_i \Delta t}$ with ϕ_i and t^0 with ϕ_0, then the explanation model of DeepLIFT will match the additive feature attribution methods (Equation (1)).

2.3.4 SHAP

SHAP (SHapley Additive exPlanations) [25] is a method to explain any machine learning model prediction. SHAP is based on the classic Shapley Values originated from game-theoretically optimal.

The intuition behind the classic Shapley values is to find a fair way for a coalition to distribute the “payout” among the “player” based on their contributions. Applying to a machine learning model, the “payout” and “player” refers to features and prediction output respectively and the Shapley values are considered as the feature importance values.

The Shapley value ϕ of the feature i from the model f_x can be obtained by the following Equation (4),

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_x(S \cup i) - f_x(S)] \tag{4}$$

where F is the set of all input features and S is a subset of the input features.

Finding the Shapley values ϕ from Equation (4) requires retraining the model. $f_x(S \cup i)$ is trained with present features where $f_x(S)$ is trained without feature i. Then all possible subsets $S \subseteq F$ were computed and weighted average all possible differences.

SHAP is the conditional expectation function from Shapley values of the original model. It provides the unique additive feature importance measure that defines the simplified inputs based on conditional expectations and satisfies three SHAP properties: local accuracy, missingness, and consistency. The local accuracy requires the summation of all feature contribution values from the explanation model to match the original model output. The missingness requires that the missing values from the original model have no impact on the explanation model. The consistency states that if a model changes and a feature has more contribution to the model, the contribution of the feature will increase or remain unchanged regardless of other features.

2.3.5 Kernel SHAP (LIME + Shapley values)

Heuristically choosing parameters for LIME in Equation (2) does not recover the Shapley values and could make LIME violate three properties of SHAP, i.e., local accuracy, missingness, and consistency.

To make LIME match the three properties of SHAP, [25] proposed a Kernel SHAP, a solution to avoid selecting LIME parameters in Equation (2) heuristically and also made Equation (2) recover the Shapley values. Kernel SHAP defines Ω, $\pi_{x'}$, and $\mathcal{L}$ for Equation (2) as follows,

$$\Omega(g) = 0 \tag{5}$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M\ choose\,|z'|)\,|z'|\,(M-|z'|)} \tag{6}$$

$$\mathcal{L}(f, g, \pi_{x'}) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{x'}(z') \tag{7}$$

where $|z'|$ is the number of non-zero elements in z'.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.3.6 DeepSHAP (DeepLIFT + Shapley values)

DeepSHAP [25, 26] is a method to explain any deep learning models by obtaining SHAP values from a given background distribution. DeepSHAP builds the connection between Shapley values and DeepLIFT. To explain the selected deep learning model, DeepLIFT needs a single background to compare with the input instance, while DeepSHAP can use single or multiple backgrounds. DeepSHAP [26] showed that from the image classification problems, DeepLIFT selected a single background as a reference result in the bias explanation. On the other hand, DeepSHAP used multiple background distributions as references which could solve the bias problem. DeepSHAP also showed that the different representative background distributions could have different explanations.

2.3.7 Summary

LIME builds a local surrogate model to explain the model's prediction while SHAP computes and retrains the model based on features. SHAP also provides a better version of LIME called Kernel SHAP, which combines the strength of both LIME and SHAP. DeepSHAP is another proposed method from the authors of SHAP to explain any deep learning model and improve the computational performance.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Chapter 3

Methods

Basically, our task is to classify the input RNA sequences into two classes: (1) long non-coding RNA (lncRNAs) and (2) messenger RNA or protein-coding RNA. Long non-coding RNA is a type of non-coding RNA, i.e., an RNA that will not be translated into protein. Still, it can function and involve in several biological mechanisms and diseases in the RNA form. In contrast, a messenger RNA or protein-coding RNA or protein-coding transcript (PCT) is an RNA that will be translated into a protein. An example of a lncRNA and a protein transcript is shown in **Figures 4, 5**, respectively.

```
>ENST00000469289.1|ENSG00000243485.5|OTTHUMG00000000959.2|OTTHUMT00000
002841.2|MIR1302-2HG-201|MIR1302-2HG|535|

TCATCAGTCCAAAGTCCAGCAGTTGTCCCTCCTGGAATCCGTTGGCTTGCCTCCGGCATTTTTG
GCCCTTGCCTTTTAGGGTTGCCAGATTAAAAGACAGGATGCCCAGCTAGTTTGAATTTTAGATA
AACAACGAATAATTTCGTAGCATAAATATGTCCCAAGCTTAGTTTGGGACATACTTATGCTAAA
AAACATTATTGGTTGTTTATCTGAGATTCAGAATTAAGCATTTTATATTTTATTTGCTGCCTCT
```

Figure 4: An example of a long non-coding RNA transcript from GENCODE in FASTA format

In this chapter, we will explain how our study will be conducted. We will use a 1D-CNN model as our training model and compare model evaluation metrics with other tools.

```
>ENST00000437963.5|ENSG00000187634.12|OTTHUMG00000040719.11|OTTHUMT000
00097862.5|SAMD11-204|SAMD11|387|UTR5:1-60|CDS:61-387|

CAGCGCTTGGGGCTCGCGGGCCGCTCCCTCCGCTCGGAAGGGAAAAGTCTGAAGACGCTTATG
TCCAAGGGGATCCTGCAGGTGCATCCTCCGATCTGCGACTGCCCGGGCTGCCGAATATCCTCC
CCGGTGAACCGGGGGCGGCTGGCAGACAAGAGGACAGTCGCCCTGCCTGCCGCCCGGAACCT
GAAGAAGGAGCGAACTCCCAGCTTCTCTGCCAGCGATGGTGACAGCGACGGGAGTGGCCCCA
```

Figure 5: An example of a protein-coding transcript from GENCODE in FASTA format

3.1 Data Preparation and Preprocessing

We obtained the human datasets for training the proposed model from GENCODE (release 32) [29] and LNCipedia (Version 5.2) [30]. The sequences dataset of GENCODE contains 48,351 lncRNAs and 100,291 PCTs. For LNCipedia, we selected only high confidence lncRNAs, which contain 107,039 lncRNAs. Since we obtained datasets from multiple sources, we further removed identical lncRNAs from GENCODE and LNCipedia by using the CD-HIS-EST-2D program [31]. We removed sequences with more than 95% similarity out, leaving a total of 72,803 lncRNAs from LNCipedia.

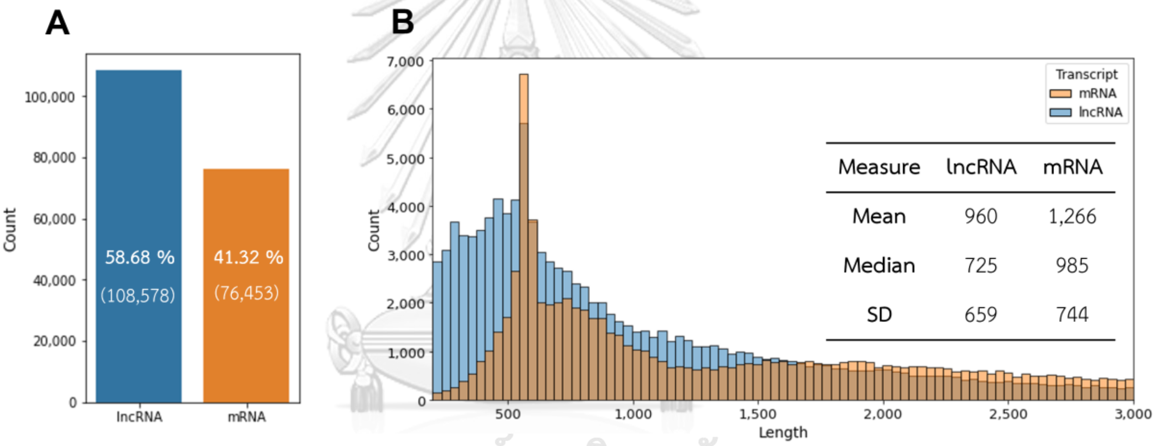
To evaluate the generalization of the model, we prepared cross-species datasets from multiple species, including mouse (GENCODE, release M23) [29], chicken, gorilla, and cow (Ensembl, release 102) [32]. The generalization datasets of mouse, chicken, gorilla, and cow contained each with an equal number of sequences from each class 32,000, 11,000, 8,000, and 8,000 sequences, respectively.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Finally, we preprocessed the sequences from all datasets by discarding sequences shorter than 200 bases and longer than 3,000 bases. We then performed one-hot encoding to encode sequences before training the proposed model. We set the lncRNAs and PCTs as the positive (1) and negative (0) class, respectively. We then stratified split the human dataset for training the model by 80% and 20% into the training and test sets. The summary of human datasets for training the proposed model is shown in **Table 4** and their distribution in **Figure 6**.

Table 4: The summary of human datasets from GENCODE and LNCipedia

Sequence	Species	Data Source	Dataset Size	< 200 bps	> 3,000 bps	Number of transcripts after cleansing
mRNA	Human	GENCODE (release 32)	100,291	374	23,464	76,453
lncRNA	Human	GENCODE (release 32)	48,351	291	3,486	44,574
lncRNA	Human	LNCipedia (version 5.2)	72,803	0	8,799	64,004



Measure	lncRNA	mRNA
Mean	960	1,266
Median	725	985
SD	659	744

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 6: (A) Distribution of target class **(B)** Distribution of the sequences' length

3.2 Model Architecture and Training Procedure

In this study, we built and designed Xlnc1DCNN architecture from scratch. We trained our model in Python3 using Tensorflow library on NVIDIA GeForce GTX 1080 Ti, Intel Xeon Silver 4112 Processor, and CentOS Linux 7. The overview of the model architecture is shown in **Table 5**.

Table 5: Hyperparameters of the proposed 1D-CNN architecture

Layer	Hyperparameter
Conv 1D	kernel size=57, stride=1
Max-Pooling	pool size=2
Dropout	p=0.3
Conv 1D	kernel size=57, stride=1
Max-Pooling	pool size=2
Dropout	p=0.3
Conv 1D	kernel size=57, stride=1
Max-Pooling	pool size=2
Dropout	p=0.3
Flatten	-
Dense	256
Dropout	p=0.5
Dense	256
Dropout	p=0.5
SoftMax	2

3.2.1 Model Hyperparameters Optimization

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

To get the above hyperparameters, we utilized 10% of the training set to perform hyperparameter tuning. We performed the grid search algorithm over the kernel and stride size of each convolutional layer, dropout rate of each dropout layer, batch size, and learning rate for model learning. The best kernel size was around 50 to 60, as shown in **Figure 7A**. After experimenting with various hyperparameters, we settled on a kernel size of 57 and a stride size of 1 for all convolutional layers. As shown in **Figure 7B**, the model performance always decreases after increasing the stride size for almost every kernel size in the search spaces.

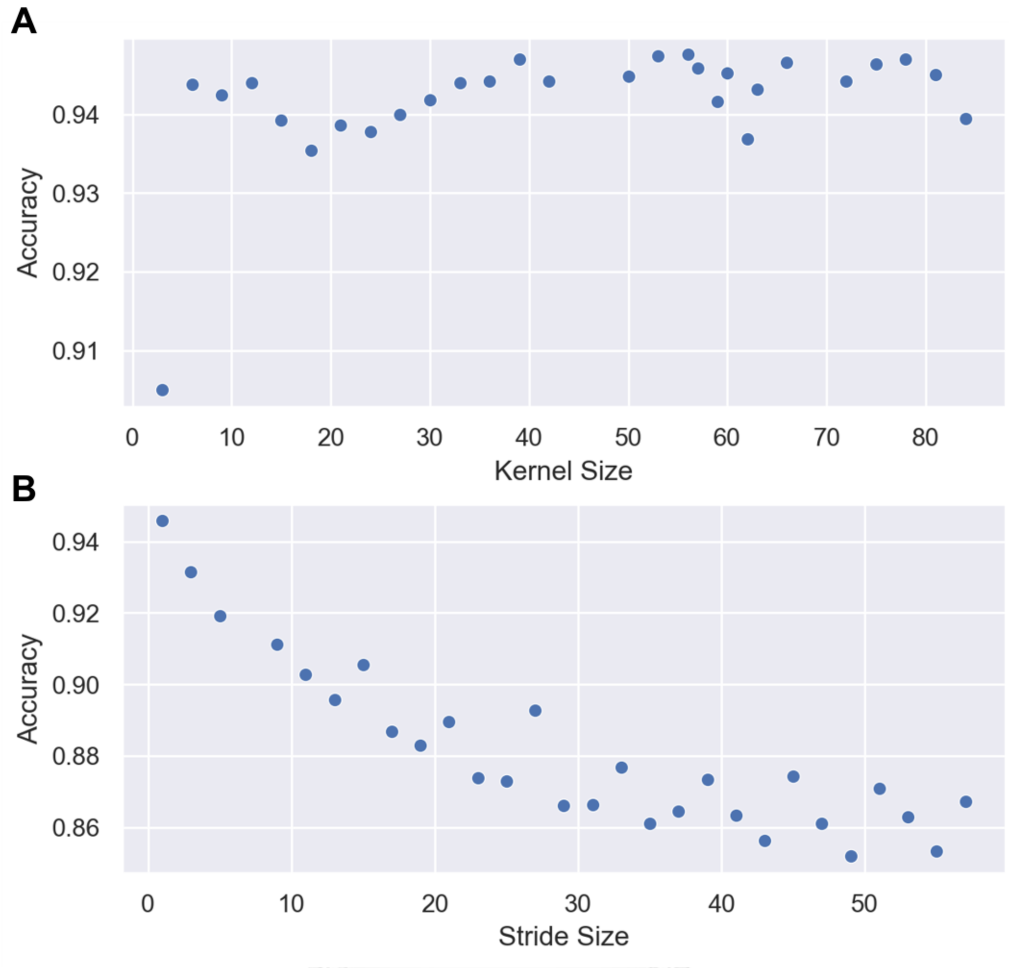


Figure 7: The model performance on the validation set. **(A)** The model performance for each kernel size **(B)** The model accuracy of a kernel size of 57 for each stride size

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.2.2 Model Hyperparameters in Details

Input data: We used 2D arrays of nucleotide sequences as the input data. The shape of the input data was (4, 3000).

Convolution Layer: We defined 120 filters, 57 kernel sizes, and 1 stride with zero-padding for each convolution layer in the model architecture. ReLu was used as the activation function of the model. We applied a weight constraint to force weights to have a magnitude below max-norm (3) to avoid overfitting.

Max-Pooling Layer: We defined 2 pool sizes and 2 strides with zero-padding for each max-pooling layer in the model architecture.

Dropout Layer: We defined a 30% dropout rate for dropout layers connected with the max-pooling layer and a 50% dropout rate for dropout layers connected with fully connected layers.

Fully Connected Layer: We defined 256 neurons for each fully connected layer in the model architecture. ReLu was also used as the activation function of the model. We also applied weight constraints below max-norm (3).

Output Layer: We defined 2 neurons and SoftMax as the activation function.

3.2.3 Details of learning

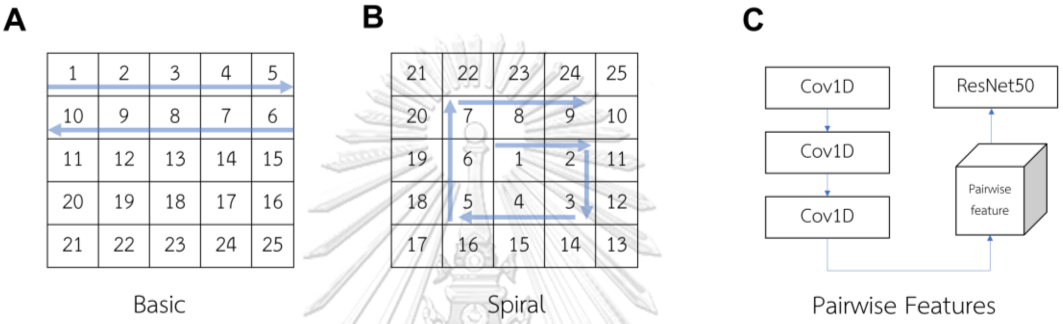
We trained our models using stochastic gradient descent as a model optimizer with a momentum of 0.9, 0.01 learning rate, 128 batch size, and 120 epochs. The training and prediction time (on the test set) was around 5 hours and 11 seconds on NVIDIA GeForce GTX 1080 Ti.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.3 Model Selection

To find the best candidate model for understanding its prediction. In this section, we compared our proposed 1D-CNN with other models presented here. All models were trained and evaluated using the human dataset.

First, we trained ResNet501D [33] with the same dataset for training our proposed 1D-CNN. We then trained ResNet50V2 (2D-CNN) using the same dataset. However, we must transform the sequences dataset to make it trainable for 2D-CNN models. The transformation methods consist of basic, spiral, and pairwise features transformation, as shown in **Figure 8**. The basic and spiral transformations transformed the sequences by rearranging them into a 2D rectangle with 60 x 50 input size. In the spiral transformations, the first nucleotide starts in the middle of the rectangle, while for the basic transformations, the first nucleotide starts at the left-most position, as shown in **Figures 8A, B**. For the pairwise features transformation, we converted the output of 1D-CNN into a 2D matrix by using the outer matrix product operation. The pairwise features transformation was inspired by the deep learning architecture for protein contact prediction [34].



1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16
21	22	23	24	25

21	22	23	24	25
20	7	8	9	10
19	6	1	2	11
18	5	4	3	12
17	16	15	14	13

Figure 8: Transformation methods for 2D-CNN (A) Basic (B) Spiral (C) Pairwise Features

Finally, we evaluated all models' performance, as shown in **Table 6**. The best model was our proposed 1D-CNN, which outperformed all other models. Thus, we will call it as Xlnc1DCNN, and we will apply the model interpretation method to obtain all insights from the model prediction result made by this model.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Table 6: Model performance of all candidate models

Model	Accuracy	Sensitivity	Specificity	Precision	F1
Proposed 1D-CNN	**94.53**	**96.22**	**92.13**	**94.55**	**95.38**
ResNet501D	87.66	95.22	76.91	85.42	90.06
ResNet50V2 (Basic)	83.70	93.95	69.14	81.22	87.12
ResNet50V2 (Spiral)	74.84	75.43	73.99	80.46	77.87
ResNet50V2 (Pairwise Features)	91.68	94.86	87.17	91.30	93.05

3.4 Model Interpretation method

We used DeepSHAP to interpret the prediction result of the proposed 1D-CNN. DeepSHAP requires input samples to be background distributions as references to approximate the SHAP values on conditional expectation. If the entire training dataset is used as a background, the expected values will be highly accurate; nevertheless, this will result in an unreasonable increase in computational cost. We then randomly selected the size of *n* background samples to observe the explanation results and to find the optimal background sample size, as shown in **Figure 9**. We found that the explanation results will remain unchanged on the background sample size greater than 200. Thus, 175 sequences were randomly selected from each class to serve as the representative background distributions. Due to the limitations of the available GPU, a total of 350 sequences were utilized.

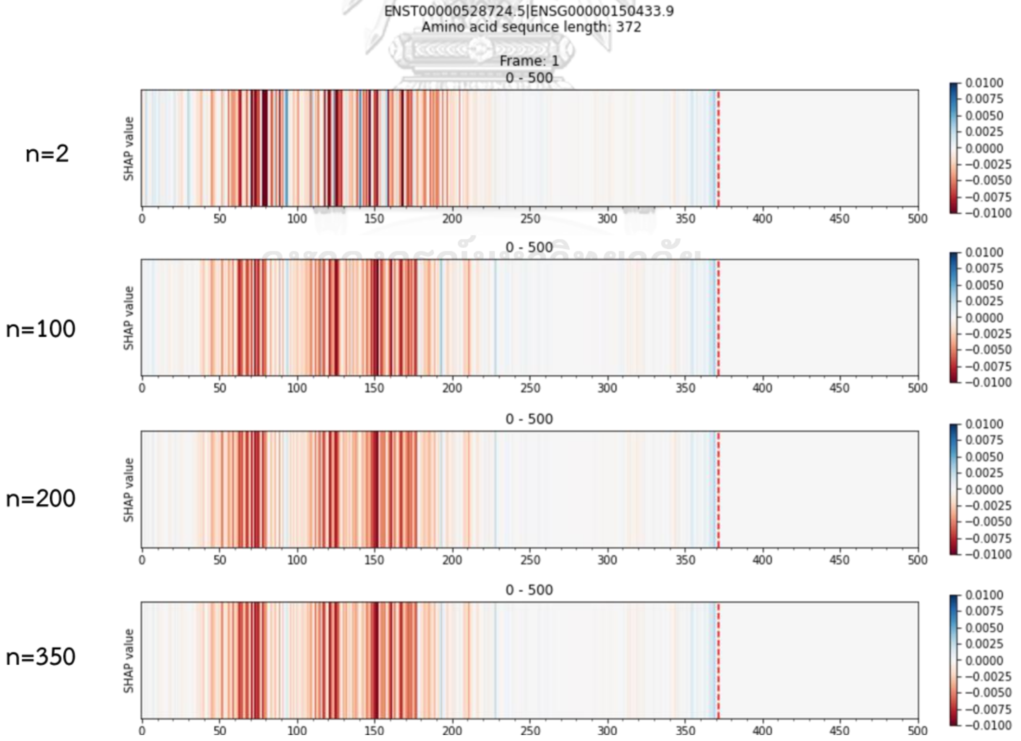


Figure 9: The explanation result from each size of the background sample

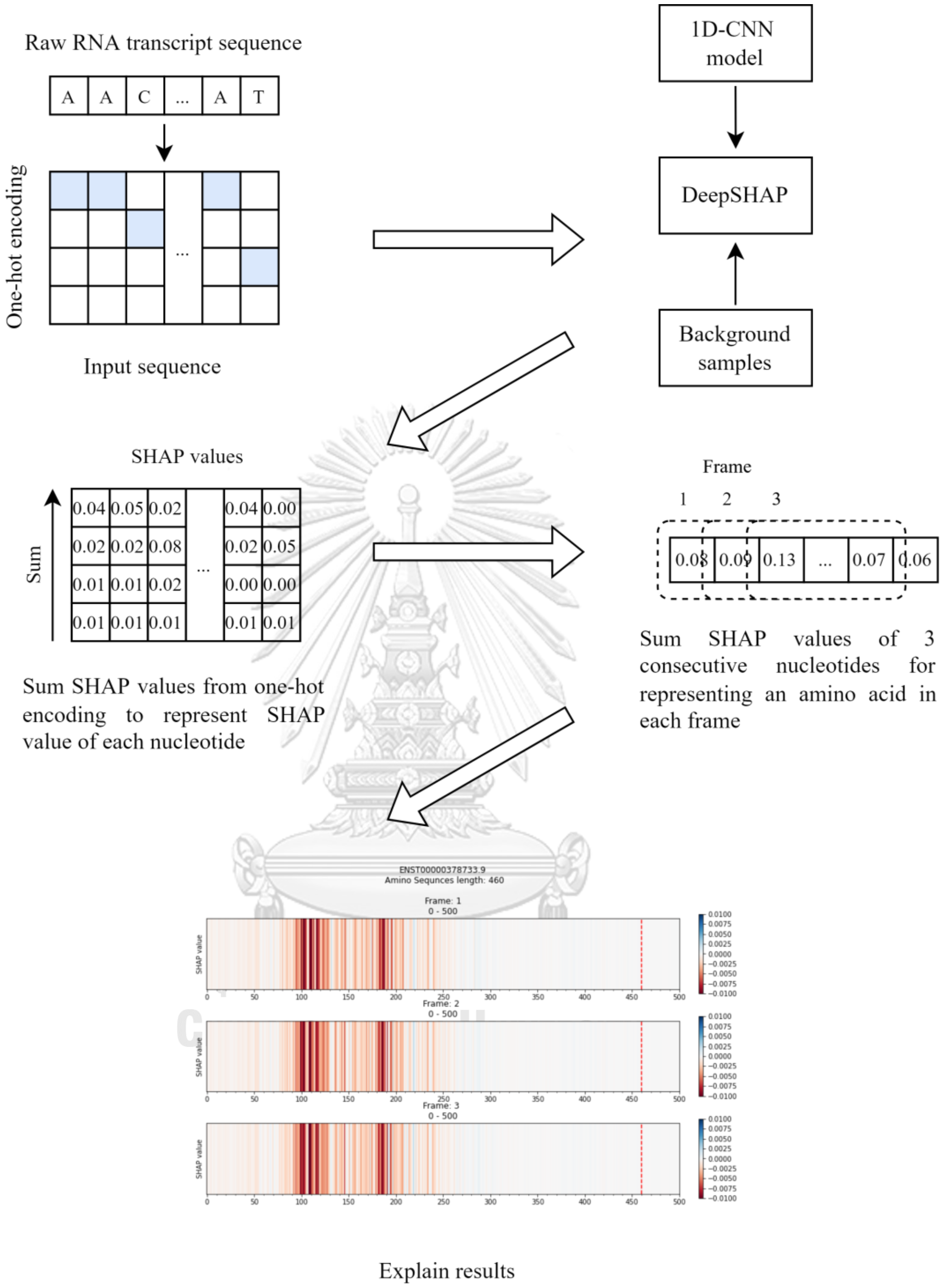


Figure 10: The process to obtain SHAP values for each amino acid

DeepSHAP generates SHAP values that represent the contribution of each training feature to the model prediction. The proposed model was trained by using solely sequences data that were encoded by one-hot encoding. Thus, we summed up SHAP values from the array of one-hot encoding to represent the SHAP value for a single nucleotide. To represent the sequence of amino acids, we further summed up the SHAP values of three consecutive nucleotides in three reading frames. Finally, we plotted a color line representing each amino acid's contribution, as shown in **Figure 10**. The red and blue colors indicate the contribution of the sequences to be classified as either an mRNA or a lncRNA.

3.5 Evaluation

3.5.1 Model Evaluation Metrics

We used the following metrics to evaluate the model performance and compare it with other tools.

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

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

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

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

$$F1 = \frac{2 \times precision \times sensitivity}{precision + sensitivity}$$

TP, true positive; *TN*, true negative; *FP*, false positive; *FN*, false negative

We then evaluate our model performance with other tools with their versions in **Table 7**.

Table 7: Software version

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Software	Version
CPC2	Web based
CNIT	Web based
PLEK	1.2
CPAT	2.0.0
FEELnc	0.1.1
RNAsamba	0.2.4
LncADeep	1.0
lncRNA_Mdeep	-

3.5.2 Interpretation Evaluation Method

We evaluated the explanation results of Xlnc1DCNN on the human test set by comparing the visualization results as described in **Chapter 3.4** with the annotation results from bioinformatics tools/databases. We assumed that Xlnc1DCNN might capture some underlying biological knowledge within sequences. Thus, we utilized TMHMM [35] to identify transmembrane helices, Pfam [36] and InterPro [37] to identify protein domains or families. From the annotation results of InterPro, we considered the following InterPro entries, which include InterPro domain, family, homologous superfamily, repeat, and sites (i.e., active site, binding site, conserved site, PTM site). We also considered the results of MobiDB (integrated within InterPro) [38] to identify intrinsically disordered regions within sequences.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter 4

Results

This chapter presents the model evaluation, interpretation, and annotation results of the proposed model.

4.1 Model Evaluation Results

We evaluated the Xlnc1DCNN performance and compared it with eight selected tools: CPAT, PLEK, FEELNC, CPC2, LncADeep, CNIT, RNAsamba, and lncRNA_Mdeep by using the human test set and cross-species datasets as described in **Chapter 3.1**. Using our human training set, we retrained tools that have a training option, including CPAT, FEELnc, and RNAsamba, and used the pre-trained models of tools that did not provide a training option, including CPC2, CNIT, and LncADeep. Although lncRNA_Mdeep and PLEK provided a training option, retraining both was very time-consuming. Thus, we decided to skip retraining and used their pre-trained models.

4.1.1 Performance Evaluation on Human Test set

The evaluation results on the human test set (**Table 8**) show that Xlnc1DCNN achieved the highest on both accuracy (94.53) and F1-Score (95.38), the second-highest precision (94.55) slightly lower than LncADeep, and the third-highest specificity (92.13) slightly lower than FEELnc and LncADeep. CPAT, CNIT, and CPC2 achieved very high sensitivity, but their specificity was much lower. FEELnc was the only tool that was trained by a machine learning algorithm (random forest) but could achieve comparable performance with other tools trained with deep learning models. FEELnc, LncADeep, RNAsamba, and lncRNA_Mdeep performed well on the average of every metric, but overall, they were still lower than our proposed model.

Table 8: Evaluation results of all tools on the human test set

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1
Xlnc1DCNN	20,895	1,204	14,087	821	**94.53**	96.22	92.13	94.55	**95.38**
CPC2	21,023	6,457	8,834	693	80.68	96.81	57.77	76.50	85.47
CNIT	21,307	3,580	11,711	409	89.22	**98.12**	76.59	85.61	91.44
PLEK	20,704	6,665	8,626	1,012	79.26	95.34	56.41	75.65	84.36
CPAT	20,646	2,597	12,694	1,070	90.09	95.07	83.02	88.83	91.84
FEELNC	20,023	1,182	14,109	1,693	92.23	92.20	92.27	94.43	93.30
RNASAMBA	20,998	1,795	13,496	718	93.21	96.69	88.26	92.12	94.35
lncRNA_Mdeep	20,813	1,799	13,492	903	92.70	95.84	88.23	92.04	93.90
LncADeep	20,232	1,113	14,178	1,484	92.98	93.17	**92.72**	**94.79**	93.97

We further analyzed the classification power of all tools, as shown in **Figure 11A**. We plotted a receiver operating characteristic curve (ROC) and calculated the area under the curve (AUC) of all tools on the human test set which Xlnc1DCNN achieved the highest AUC (0.9825). We also calculated the accuracy of all tools on each range of sequence lengths as shown in **Figure 11B**, and Xlnc1DCNN also outperformed all tools on any range of sequences on the human test set.

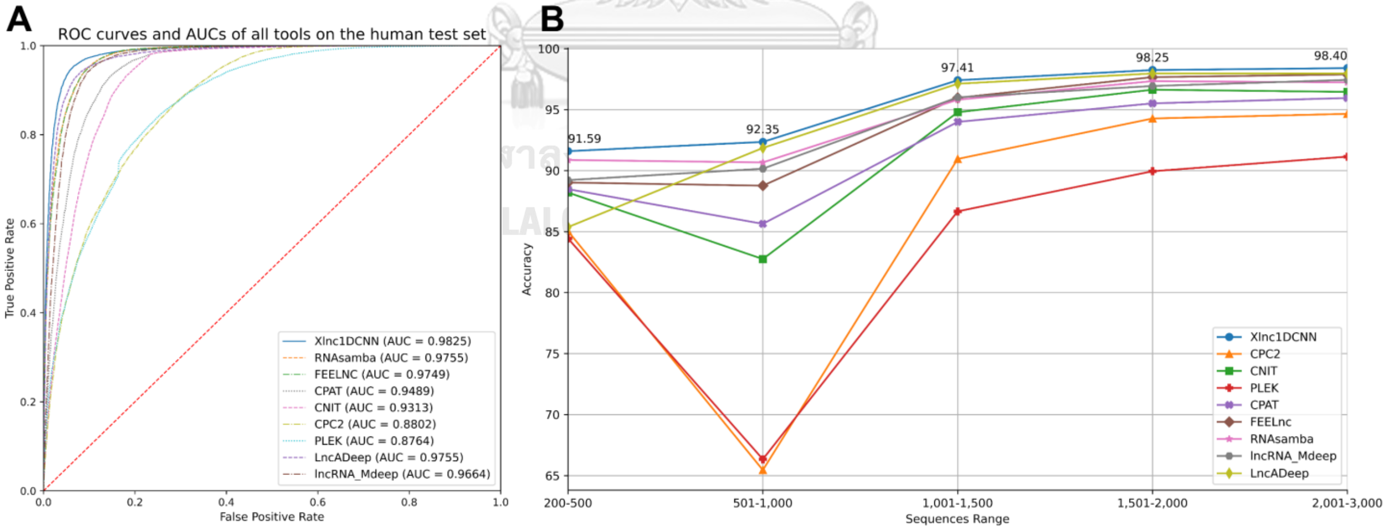


Figure 11: (A) ROC curve of all tools and their AUC on the human test set **(B)** Accuracy of all tools for any range of sequence lengths on the human test set

4.1.2 Performance Evaluation on Cross-Species Datasets

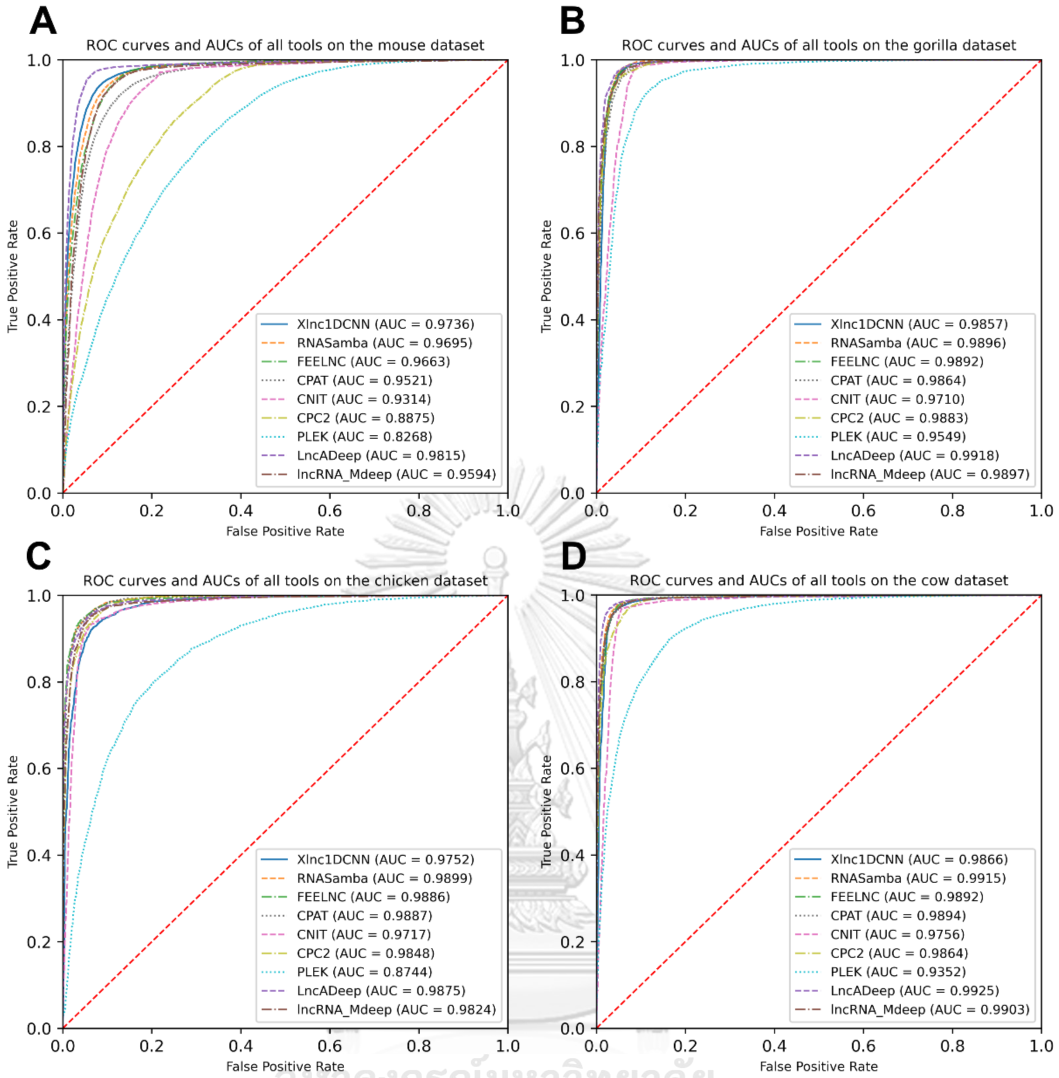
In the previous section, we showed that Xlnc1DCNN could achieve high performance compared with other tools on the human test set. Thus, we also need to evaluate the generalization of the model to verify that it was not overfitting on the human dataset. We used cross-species datasets (mouse, gorilla, chicken, and cow) as described in **Chapter 3.1** to evaluate the generalization of Xlnc1DCNN. **Table 9** shows that Xlnc1DCNN achieved the highest accuracy on gorilla dataset alongside RNAsamba and the second highest accuracy on mouse dataset. While on both mouse and cow datasets, LncADeep achieved the highest accuracy. Other evaluation metrics are shown in **Appendices A1-A4**.

Table 9: Accuracy of all models on cross-species datasets

Model	Mouse	Gorilla	Chicken	Cow
Xlnc1DCNN	92.58	**96.06**	92.35	95.92
CPC2	80.06	94.96	93.51	94.48
CNIT	87.68	94.00	92.94	95.18
PLEK	73.62	89.53	79.54	86.22
CPAT	89.46	95.1	93.70	95.52
FEELnc	90.51	94.8	92.75	93.97
RNAsamba	91.91	**96.06**	**93.98**	96.39
LncADeep	**94.95**	96.05	93.46	**96.70**
lncRNA_Mdeep	91.38	95.58	92.59	95.63

Figure 12 shows that the ROC curves and AUCs of Xlnc1DCNN for cross-species datasets are comparable to other methods. LncADeep had the best generalization performance on cross-species datasets based on AUCs. In summary, the evaluation results of all species showed that Xlnc1DCNN has the generalization for classifying lncRNAs and mRNAs for other species.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 12: Receiver operating characteristic curves and AUCs of nine models on the datasets of **(A)** mouse **(B)** gorilla **(C)** cow and **(D)** chicken

4.2 Model Interpretation Results

Xlnc1DCNN was built based on the CNN model, which is also known as one of the deep learning techniques that can learn complex patterns within the data. We can presume that our trained 1D-CNN captured some patterns within sequences that could be used to classify between lncRNAs and mRNAs. To understand these patterns. In this section, we used DeepSHAP to explain what the model has learned by plotting the contribution of each nucleotide for the entire sequence, as we described in **Chapter 3.4.**

In the following subsections, we will present the explanation results of Xlnc1DCNN for true positive, true negative, false positive, and false negative sequences, respectively.

4.2.1 True Positive Sequences

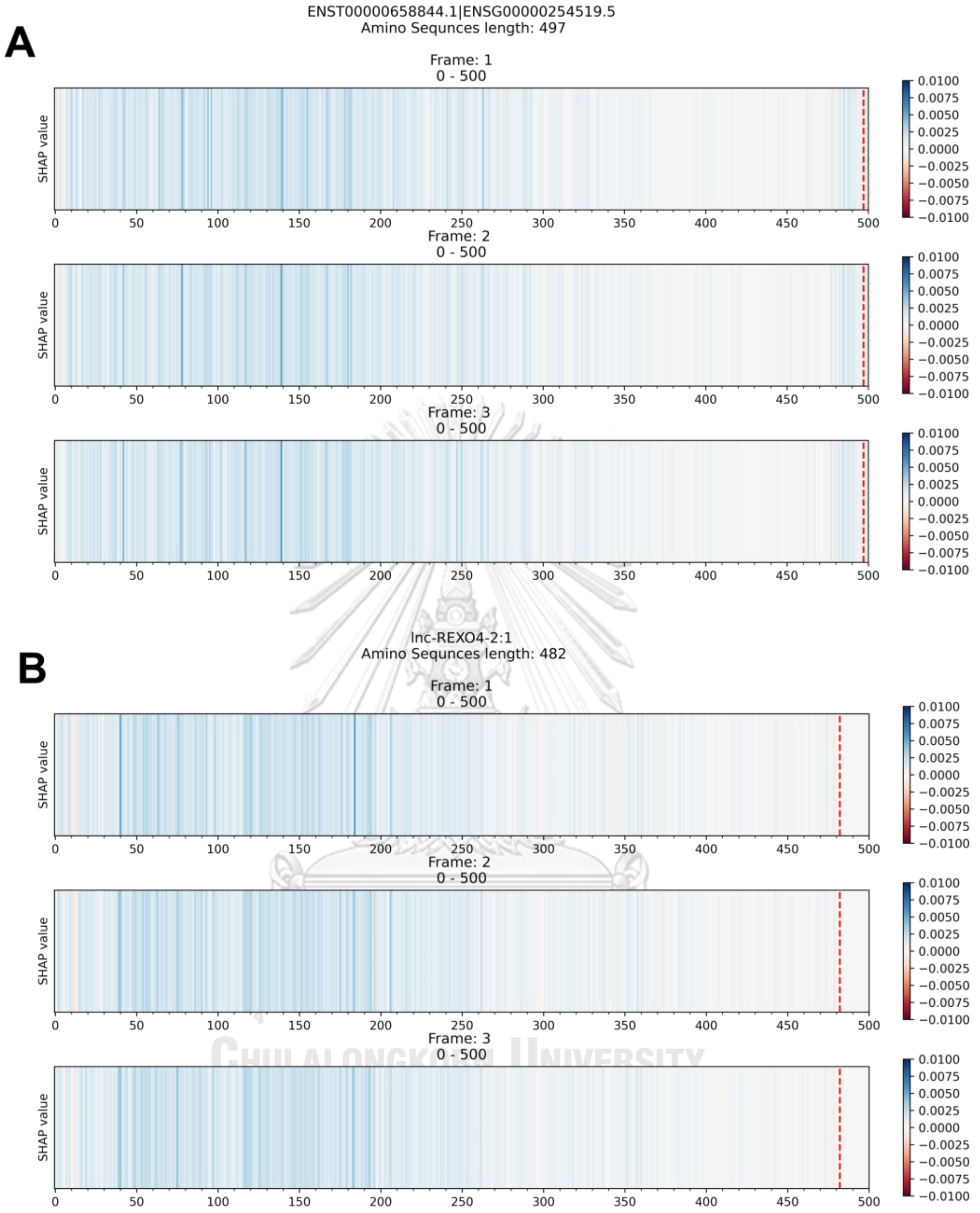
True positive sequences (TPs) are lncRNAs that are correctly classified as lncRNA. The explanation results highlighted the important regions that contributed to the correct classification as lncRNA in blue color. We summarized the patterns that we found from the explanation results of TPs into four categories.

In the first category, Xlnc1DCNN highlighted weak signals across the entire sequence, as shown in **Figures 13A, B**. The explanation results of the ENST00000658844.1 and lnc-REXO4-2:1 suggested that Xlnc1DCNN classified a transcript sequence as a lncRNA if any important regions or distinctive patterns within the sequence were not captured. The annotation result of InterPro also couldn't identify any entries.

In the second category, Xlnc1DCNN particularly highlighted important regions in some parts of the sequence, as shown in **Figure 14A**. To understand these important regions, we utilized the bioinformatic tools, as described in **Chapter 3.5**. The prediction of TMHMM (**Figure 14B**) shows that these regions were predicted as a transmembrane helix. Thus, the explanation results of the lnc-NXNL1-3:11 have shown that Xlnc1DCNN has learnt transmembrane helices as one of the patterns that contributed to the prediction as lncRNA.

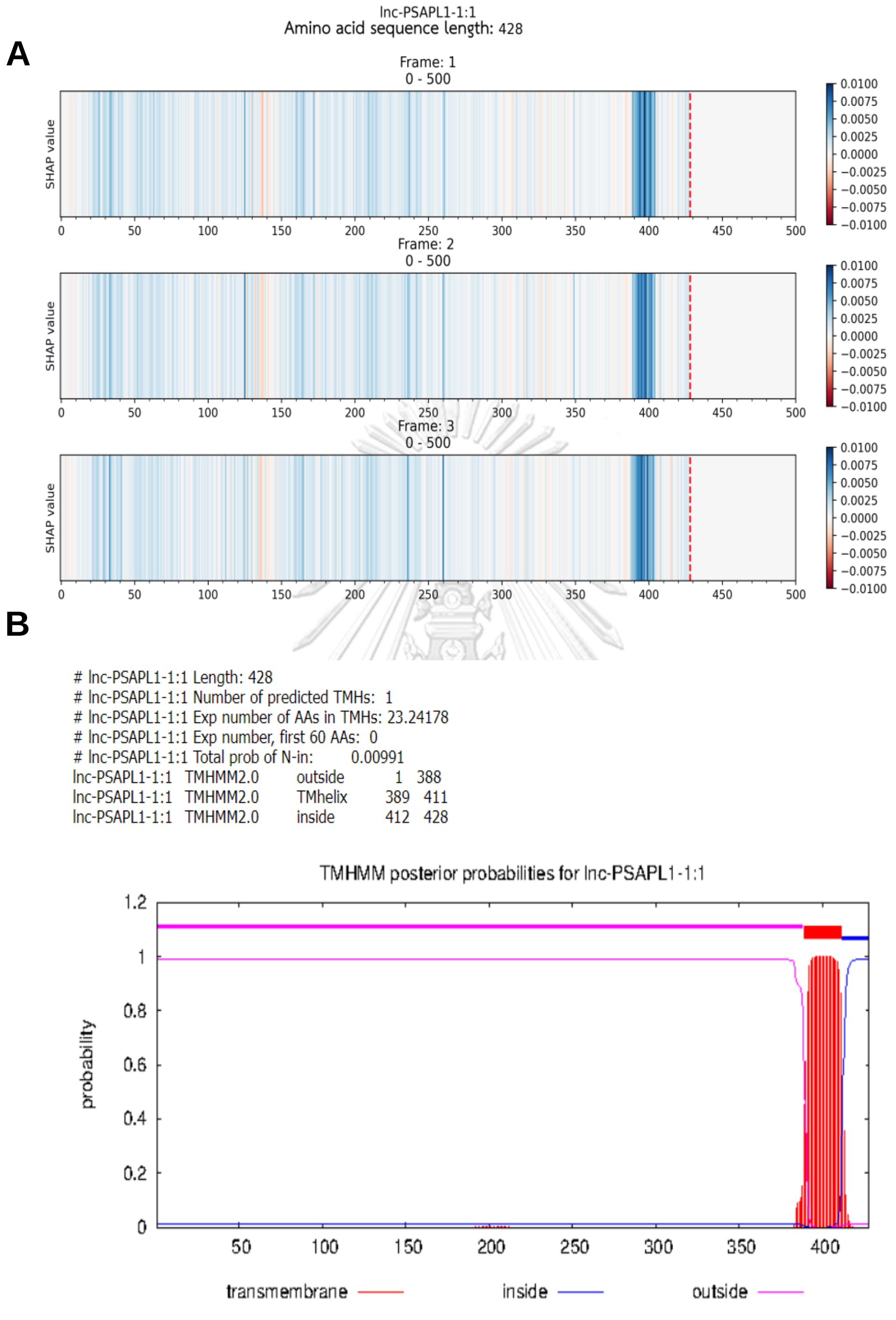
In the third category, Xlnc1DCNN highlighted important regions, particularly in the front part of the sequence, as shown in **Figure 15A.** The explanation result of the ENST00000663782.1 highlighted important regions which corresponded with the identification by InterPro as signal N peptide terminal region as shown in **Figure 15B**.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



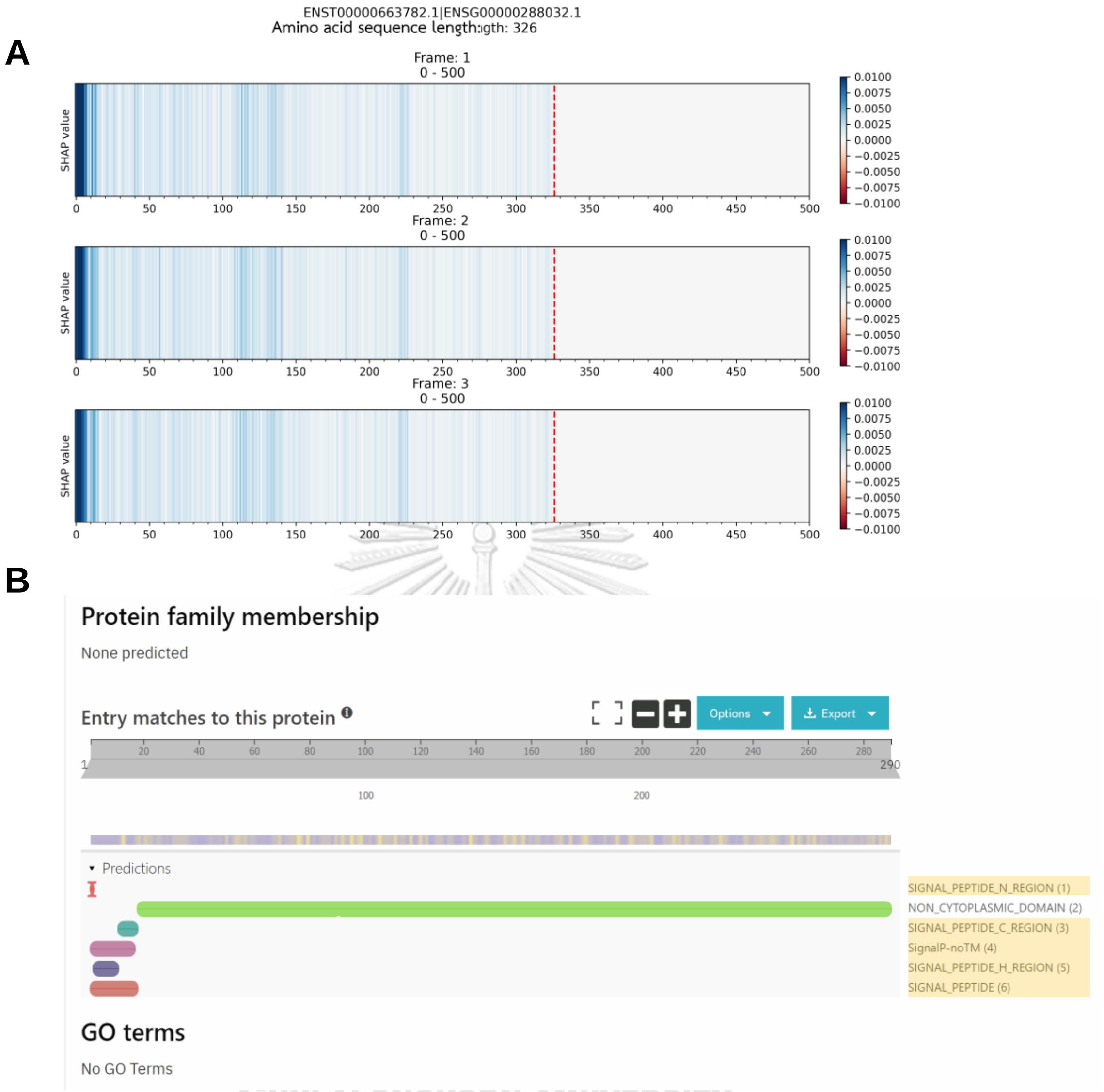
CHULALONGKORN UNIVERSITY

Figure13: Explanation results of Xlnc1DCNN highlighted weak signals on TP sequences **(A)** ENST00000658844.1, a lncRNA sequence obtained from GENCODE and **(B)** lnc-REXO4-2:1, a lncRNA sequence obtained from LNCipedia.



```
# lnc-PSAPL1-1:1 Length: 428
# lnc-PSAPL1-1:1 Number of predicted TMHs:  1
# lnc-PSAPL1-1:1 Exp number of AAs in TMHs: 23.24178
# lnc-PSAPL1-1:1 Exp number, first 60 AAs:  0
# lnc-PSAPL1-1:1 Total prob of N-in:        0.00991
lnc-PSAPL1-1:1  TMHMM2.0       outside       1   388
lnc-PSAPL1-1:1  TMHMM2.0       TMhelix     389   411
lnc-PSAPL1-1:1  TMHMM2.0       inside      412   428
```

Figure 14: (A) Explanation result of Xlnc1DCNN and **(B)** the prediction result of the transmembrane helices by TMHMM program on the true positive sequence, lnc-NXNL1-3:11 which corresponds to the explanation result of Xlnc1DCNN.



Protein family membership

None predicted

Entry matches to this protein

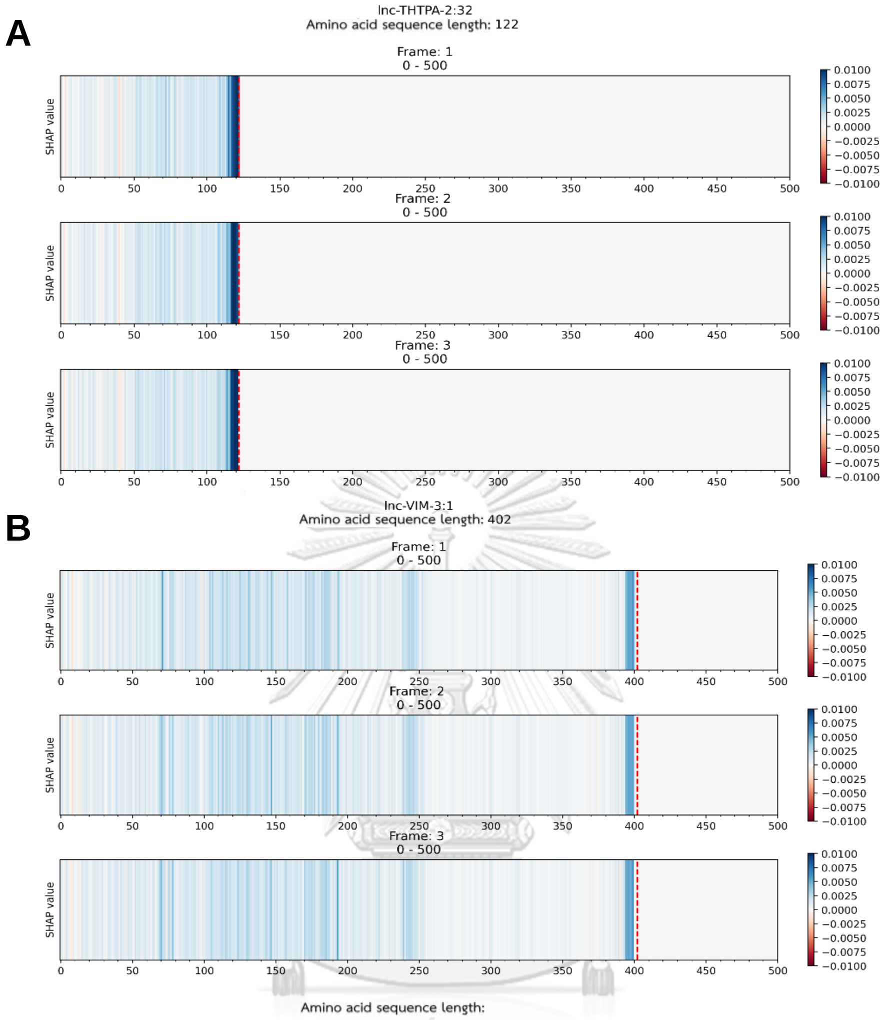
GO terms

No GO Terms

CHULALONGKORN UNIVERSITY

Figure 15: (A) Explanation result of Xlnc1DCNN and **(B)** the signal peptide identified by InterPro on the true positive sequence, ENST00000663782.1 obtained from GENCODE.

In the last category, Xlnc1DCNN particularly highlighted important regions in some parts of the sequence. However, these regions do not correspond with any identification results from InterPro, Pfam, and TMHMM. **Figure 16** shows the explanation results of the lnc-THTPA-2:32 and lnc-VIM-3:1 and these important regions do not correspond to any InterPro entries.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 16: The explanation results of Xlnc1DCNN on true positive sequences of the **(A)** lnc-THTPA-2:32, **(B)** lnc-VIM-3:1 obtained from LNCipedia.

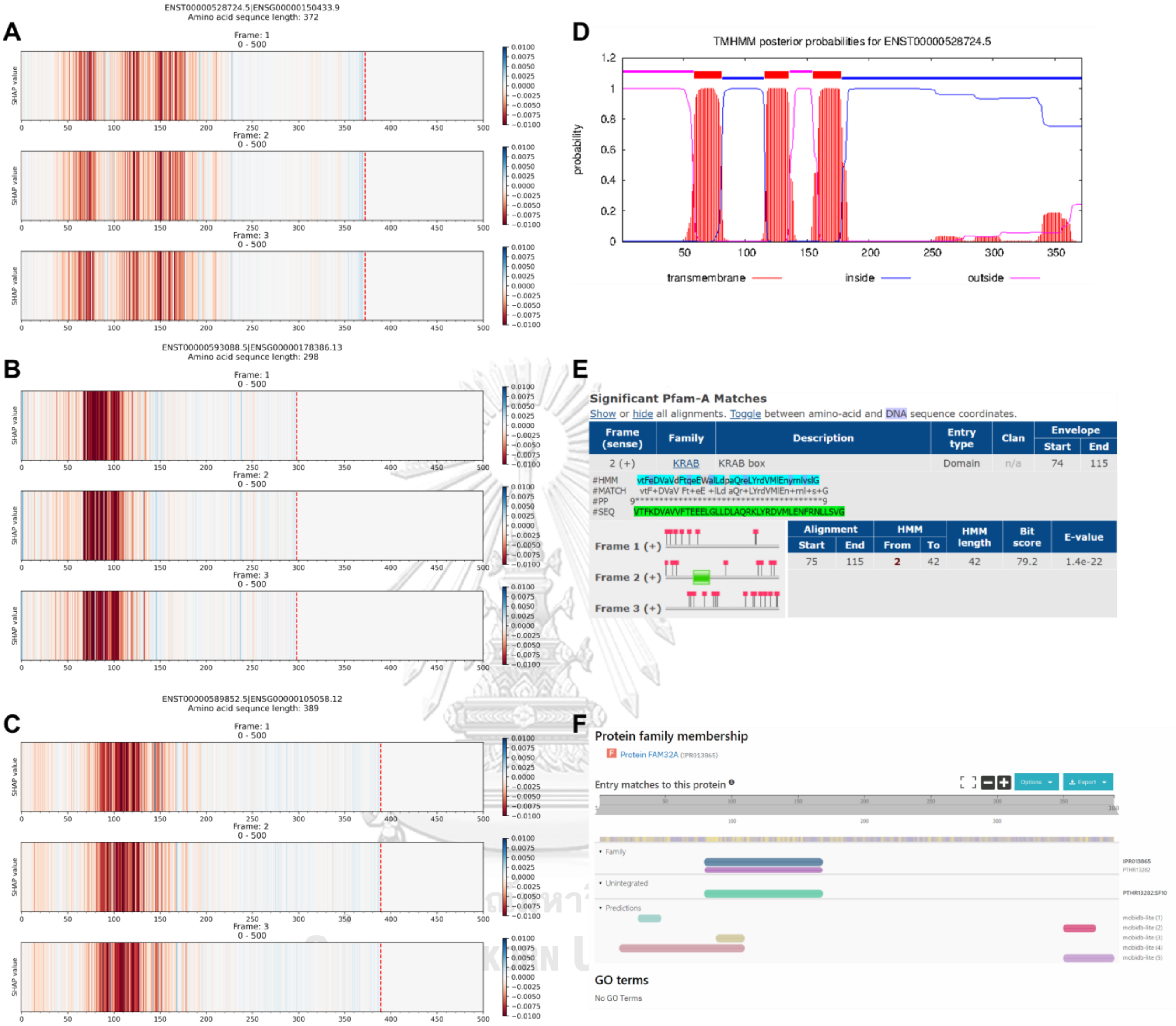
4.2.2 True Negative Sequences

True negative sequences (TNs) are mRNAs that are correctly classified as mRNA. The important regions of the explanation results were highlighted in red, indicating mRNAs' prediction contribution, as shown in **Figures 17A-C**. We found that most of the import regions of TNs were found with protein domains and/or families, as shown in **Figures 17D-E. Figure 17D**, the ENST00000528724.5 transcript, shows the prediction result of transmembrane helix regions by TMHMM, which corresponds to the important regions highlighted by the Xlnc1DCNN in **Figure 17A. Figure 17B**, the ENST00000593088.5 transcript, shows the important region highlighted by Xlnc1DCNN which overlapped the identification result of the KRAB box (Krüppel associated box) by Pfam as shown in **Figure 17E**. **Figure 17F**, the ENST00000589852.5 transcript, shows the identification result of the FAM32A family (family with sequence similarity 32 member A) by InterPro, which also corresponds to the important region highlighted by the Xlnc1DCNN as shown **Figure 17C**. This transcript has been related to an ovarian tumor gene [39].

4.2.3 False Positive Sequences

False positive sequences (FPs) are mRNA transcript sequences that are incorrectly classified as lncRNAs. Unlike the explanation results of TNs, the explanation results of FPs had a similar pattern to the explanation results of TPs, which did not contain any important regions in red color. **Figure 18A**, the ENST00000408930.6 transcript, shows the explanation result which did not highlight any important regions that contributed to the prediction as an mRNA in red. The identification results of Pfam and InteroPro also couldn’t identify any protein domains or families, as shown in **Figures 18B, C.** We then investigated the ENST00000408930.6 across all databases. We found that while the Ensembl database reports the ENST00000408930.6 as a protein-coding transcript of the HEPN1 (ENSG00000221932) gene, the Gene database at NCBI reports HEPN1 as the ncRNA gene (https://www.ncbi.nlm.nih.gov/gene/641654) and the RefSeq database reports the NR_170124.1 (ENST00000408930.6) as a long non-coding RNA (https://www.ncbi.nlm.nih.gov/nuccore/NR_170124.1). The prediction results of other top tools (FEELnc, LncADeep, RNAsamba, lncRNA_Mdeep) also classified this sequence as lncRNA. This sequence illustrates an instance of conflicting annotations among public databases, which impact the performance of the model and its explanation.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



Significant Pfam-A Matches

Show or hide all alignments. Toggle between amino-acid and DNA sequence coordinates.

Frame (sense)	Family	Description	Entry type	Clan	Envelope	
					Start	End
2 (+)	KRAB	KRAB box	Domain	n/a	74	115

```
#HMM      vtFeDVaVdFtqeEWalLdpaQreLYrdVMlEnyrnlvslG
#MATCH     vtF+DVaV Ft+eE +lLd aQr+LYrdVMlEn+rnl+s+G
#PP      9****************************************9
#SEQ     VTFKDVAVVFTEEELGLLDLAQRKLYRDVMLENFRNLLSVG
```

Alignment		HMM		HMM length	Bit score	E-value
Start	End	From	To			
75	115	2	42	42	79.2	1.4e-22

Protein family membership

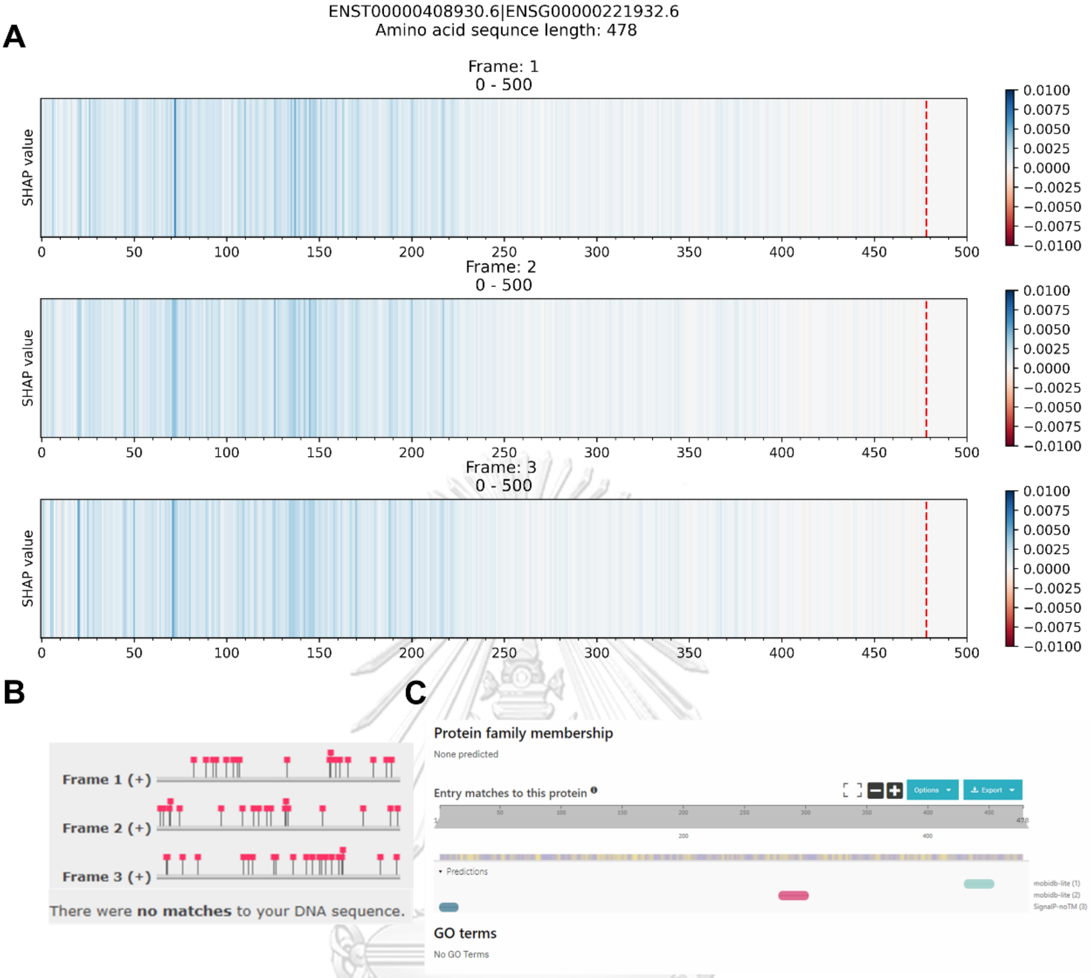
Protein FAM32A (IPR013865)

Entry matches to this protein

GO terms

No GO Terms

Figure 17: Comparison between the explanation results of Xlnc1DCNN on TN sequences **(A)** ENST00000528724.5, **(B)** ENST00000593088.5, and **(C)** ENST00000589852.5 protein-coding transcripts and **(D)** the prediction result of TMHMM program on the ENST00000528724.5 **(E)** the KRAB (Krüppel associated box) domain identified by Pfam within the ENST00000593088.5 and **(F)** the FAM32A family identified by InterPro within the ENST00000589852.5 transcripts.



There were **no matches** to your DNA sequence.

None predicted

No GO Terms

Figure 18: Comparison between **(A)** the explanation result of Xlnc1DCNN on the ENST00000408930.6 protein-coding transcript, predicted as a lncRNA **(B)** the identification result from Pfam and **(C)** the identification result from InterPro.

4.2.4 False Negative Sequences

False negative sequences (FNs) are lncRNAs that incorrectly classified as mRNAs. **Figures 19A, B**, the LNC-SIGIRR-2:1 and ENST00000616537.4, show the highlighted important regions that contributed to misclassifying as mRNA transcripts. The identification results of InterPro identified protein families of Anoctamin and Taxilin, respectively, as shown in **Figures 19C, D**. In contrast to typical lncRNAs, these FNs contain protein domains/families regions.

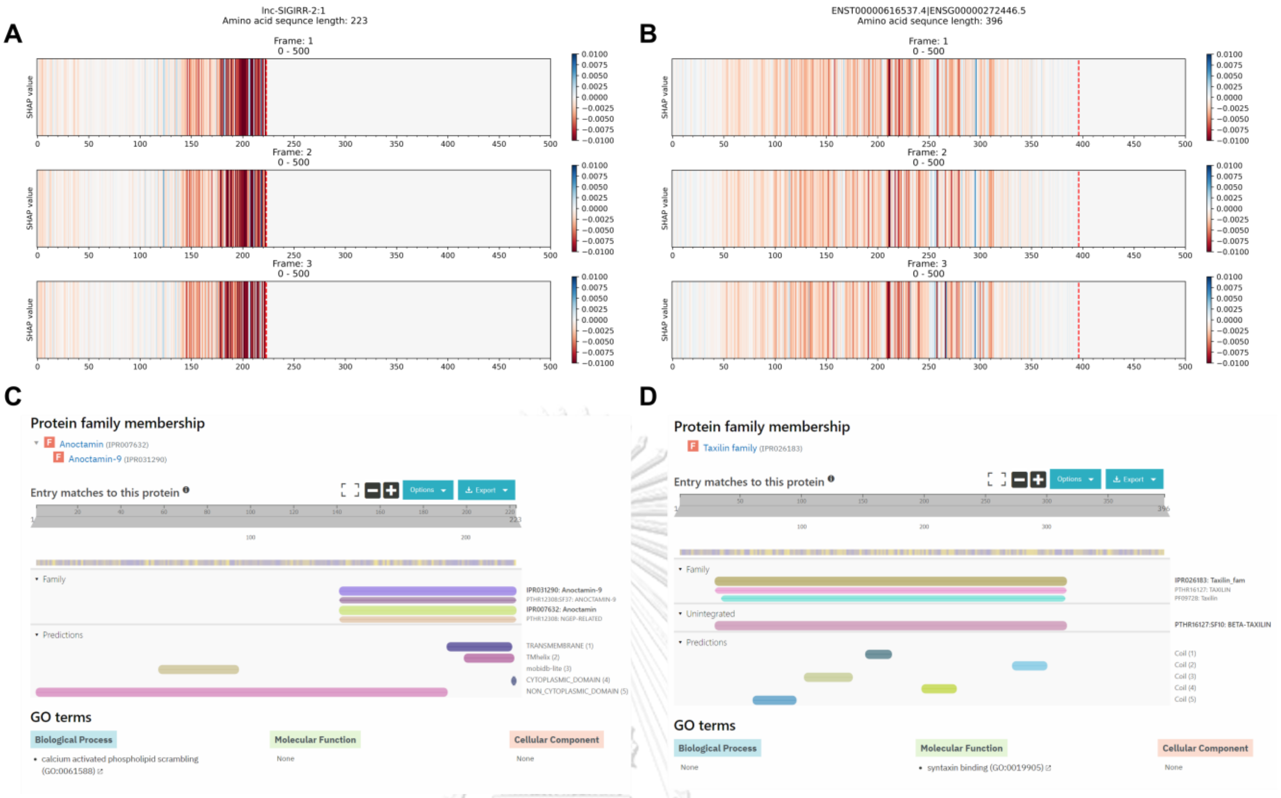


Figure 19: Comparison between the explanation result of Xlnc1DCNN on the long non-coding RNA transcripts **(A)** lnc-SIGIRR-2:1 and **(B)** ENST00000616537.4, predicted as mRNAs **(C)** the Anoctamin family within the lnc-SIGIRR-2:1 transcript and **(D)** the Taxilin family within the ENST00000616537.4 transcript identified by InterPro.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.3 Sequences Annotation results

To reassure our findings from the explanation results in the previous section. In this section, we summarized the annotation results of all sequences on the test by using the result of InterPro entries as described in **Chapter 3.5**. The summary is shown in **Table 10**.

Table 10: Summary of test set sequences annotated with InterPro entries

Metrics	Amount	Found with InterPro Entries	Found without InterPro Entries	Contain IDRs without InterPro Entries	Contain Transmembrane Helixes without InterPro Entries
TP	20,895	1,692 (8.10%)	19,203 (91.9%)	9,833 (47.06%)	7,713 (36.91%)
TN	14,087	13,085 (92.89%)	1,002 (7.11%)	822 (5.84%)	289 (2.05%)
FP	1,204	704 (58.47%)	500 (41.53%)	359 (29.82%)	161 (13.37%)
FN	821	463 (56.39%)	358 (43.61%)	264 (32.16%)	104 (12.67%)
All missed FP	344	93 (27.03%)	251 (72.97%)	164 (47.67%)	94 (27.33%)
All missed FN	105	90 (85.71%)	15 (14.92%)	5 (4.76%)	4 (3.81%)

We first considered the annotation results of true positive sequences. From the total of 20,895 TPs, only 8.10% (1,692/20,895) TPs contained InterPro entries. However, top protein domains and families of these entries on the test set were found in only a few TNs (≤5) (**Appendices A5-A6**). The 47.06% (9,833/20,895) TPs were found with only intrinsically disordered regions (IDRs), and 36.91% (11,490/20,895) TPs were found with transmembrane helices identified by TMHMM without any InterPro entries. These annotation results showed that most lncRNAs did not contain any InterPro entries.

CHULALONGKORN UNIVERSITY

As shown in **Figure 15**, the explanation results for true negative sequences indicate that our model could capture the regions in the transcript sequences that represent protein domains or families. From a total of 14,087 TNs, 92.86% (13,079/14,087) were found with InterPro entries, 5.84% (882/14,087) were found with only IDRs, and 2.05% (289/14,087) TNs contained only transmembrane helices without any InterPro entries. Hence, the majority of TNs contained InterPro entries and our model could correctly classify most of the input mRNAs.

The explanation results of false positive sequences typically do not contain the important regions (red color) that contributed to the model prediction as mRNAs. Of the total 1,204 FPs, 42.53% (500/1204) FPs were found without any InterPro entries, 29.81% (359/1204) FPs were found with only IDRs, and 13.37% (161/1204) FPs were found with transmembrane helices without any InterPro entries.

The 56.39% (463/821) FNs contained InterPro entries, and the explanation results also highlighted the important regions in red, as shown in **Figure 19**. In addition, 32.16% (264/821) and 12.67% (104/821) FNs were found with IDRs and transmembrane helices, respectively.

Finally, we could summarize the annotation results of InterPro entries from TP, TN, FP, and FN as shown in **Table 10**. Most of the TPs were found without any InterPro entries, while TNs, most of which were found with InterPro entries that correspond with the explanation results of Xlnc1DCNN. The number of TPs annotated with only transmembrane helices or IDRs also highlighted these regions' contributions to predicting the sequences as lncRNAs.

The 58.47% (704/1,204) and 43.61% (358/821) annotated FPs and FNs with and without InterPro entries indicated the limitations of Xlnc1DCNN. We also analyzed top tools (Xlnc1DCNN, FEELnc, LncADeep, RNAsamba, lncRNA_Mdeep) misclassified FPs and FNs. The 27.03% (93/344) and 14.92% (15/105) annotated FPs and FNs with and without InterPro entries misclassified by all top tools suggested complicated sequences to classify. In addition, the misclassification of 72.97% (251/344) and 85.71% (90/105) annotated FPs and FNs without and with InterPro entries by all top tools suggests the limitations of all top tools or inconsistent annotations among public databases.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter 5

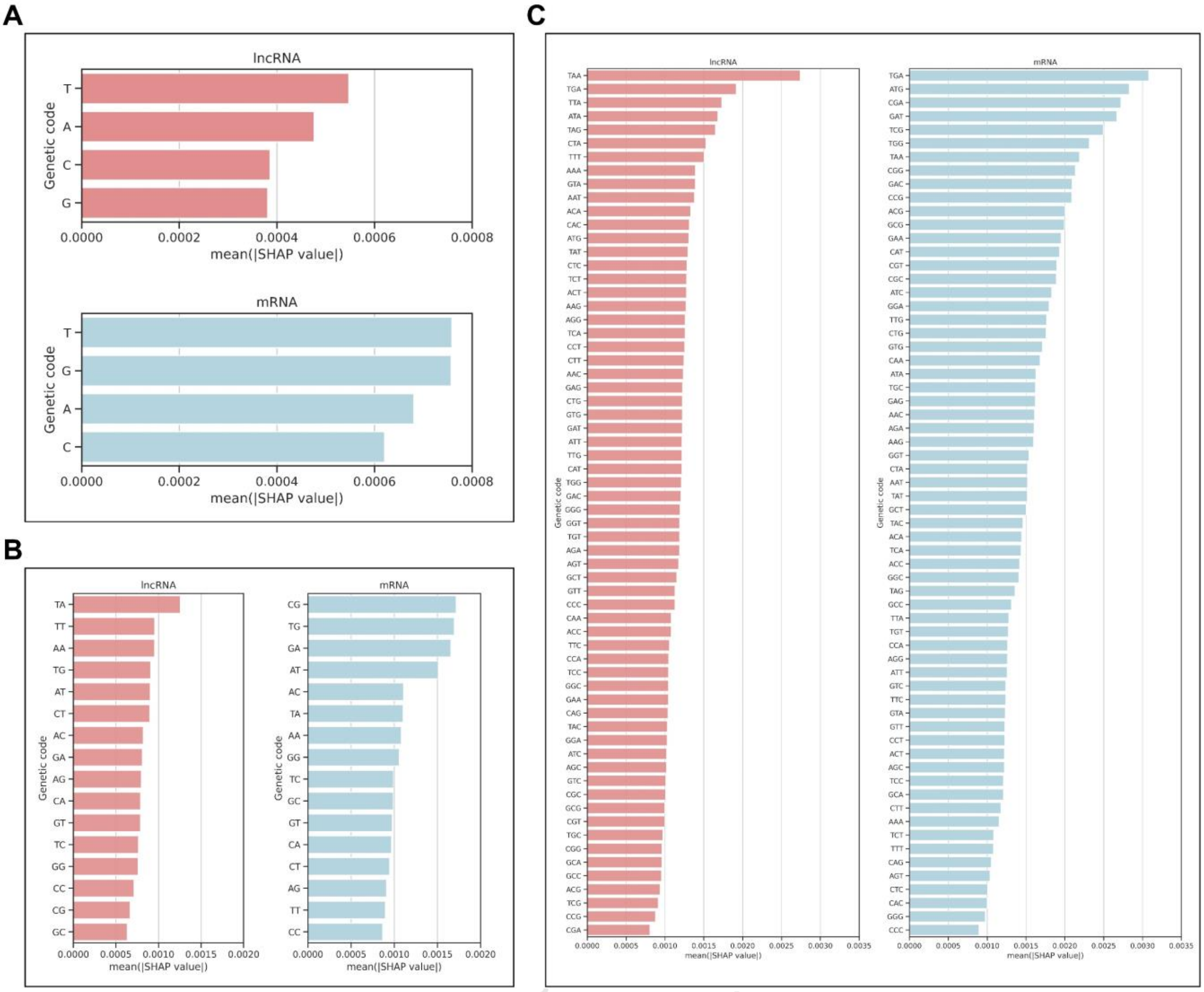
Discussion

We have shown that the explanation results can help understand how our model learned to differentiate between lncRNAs and mRNAs. Furthermore, these findings from the explanation results are also consistent with other studies; for example, [40, 41] found a transmembrane helix within lncRNAs, and [42] reported hidden peptides encoded inside non-coding RNAs, which is consistent with the highlighted regions perceived by Xlnc1DCNN for classifying lncRNAs from mRNAs.

Besides, we also investigated how the single nucleotide, dinucleotide, and trinucleotide (codon) contributed to the prediction results by plotting their mean of absolute SHAP values, as shown in **Figure 20**. The higher the mean of absolute SHAP values, the more significant contribution of that genetic code. The stop codons TAA, TGA, and TTA, were the top three codons with the highest contribution to lncRNA prediction, while the stop codon (TGA), the start codon (ATG), and arginine (CGA) were the top three codons for mRNA prediction. The CG was the top dinucleotides contributing to the mRNA prediction, consistent with [43].

According to findings from more recent research, some putative lncRNAs contain a short open reading frame (sORF) [44]. We also tried to analyze the relationship between lncRNAs and sORF using the explanation results of Xlnc1DCNN. We randomly selected some false negative sequences and verified whether they contained sORFs by using MetemORF [45]. Some of the false negative sequences were found with sORFs. However, the important regions highlighted by the explanation results and the reported regions of sORFs were inconsistent.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 20: Mean of absolute SHAP values for **(A)** single nucleotide **(B)** dinucleotide and **(C)** trinucleotide, indicating the impact of each genetic code on the model prediction as lncRNA or mRNA.

Conclusion

This thesis proposed Xlnc1DCNN, a simple but effective 1D-CNN model for classifying lncRNA and mRNAs (protein-coding transcripts) integrated with prediction explanation results. Furthermore, we have shown that using 1D-CNN as a feature extractor instead of applying traditional feature extraction methods can result in a more accurate prediction than other available tools.

The explanation results revealed several insights about how Xlnc1DCNN learned to differentiate the lncRNAs from mRNAs. The recent findings of transmembrane microproteins within lncRNAs agreed with the transmembrane helices area highlighted by the explanation results of multiple true positive lncRNAs while several misclassified lncRNAs contained protein domains or families in Pfam and/or InterPro. In addition, several misclassified mRNAs were disordered proteins that did not contain any highlighted regions in the explanation results. These findings bring insights into the complexities of long non-coding RNAs and suggest the necessity of regular evaluations of cross-referenced gene annotations among public databases. Xlnc1DCNN, together with all explanation results, are publicly available at https://github.com/cucpbioinfo/Xlnc1DCNN.

Future Work

Although our model could outperform other long non-coding identification tools using a simple 1D-CNN architecture, different approaches could be used to improve the model performance further. For example, set the dilation rate to expand the convolution kernel size instead of increasing the kernel size, experiment with other deep learning models, or use different hyperparameter tuning algorithms instead of the grid search algorithm.

LSTM (Long Short-Term Memory), RNN (Recurrent Neural Networks), and GRU (Gated Recurrent Units) could be a candidate model for improving the model performance if the order of the nucleotide sequences is important. The grid search algorithm could be replaced with NNI (Neural Network Intelligence) [46] or OPTUNA [47] which is one of the popular hyperparameter tuning algorithms for searching the optimal hyperparameter and neural architecture.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

REFERENCES

1. Marchese, F.P., I. Raimondi, and M. Huarte, *The multidimensional mechanisms of long noncoding RNA function.* Genome Biology, 2017. **18**(1): p. 206.
2. Rinn, J.L. and H.Y. Chang, *Genome Regulation by Long Noncoding RNAs.* Annual Review of Biochemistry, 2012. **81**(1): p. 145-166.
3. Statello, L., et al., *Gene regulation by long non-coding RNAs and its biological functions.* Nature Reviews Molecular Cell Biology, 2021. **22**(2): p. 96-118.
4. Jin, K.-T., et al., *Roles of lncRNAs in cancer: Focusing on angiogenesis.* Life Sciences, 2020. **252**: p. 117647.
5. Fang, Y. and M.J. Fullwood, *Roles, Functions, and Mechanisms of Long Non-coding RNAs in Cancer.* Genomics, Proteomics & Bioinformatics, 2016. **14**(1): p. 42-54.
6. Morán, I., et al., *Human β cell transcriptome analysis uncovers lncRNAs that are tissue-specific, dynamically regulated, and abnormally expressed in type 2 diabetes.* Cell metabolism, 2012. **16**(4): p. 435-448.
7. Chan, J.J. and Y. Tay, *Noncoding RNA:RNA Regulatory Networks in Cancer.* International Journal of Molecular Sciences, 2018. **19**(5): p. 1310.
8. Stark, R., M. Grzelak, and J. Hadfield, *RNA sequencing: the teenage years.* Nature Reviews Genetics, 2019. **20**(11): p. 631-656.
9. Wang, Z., M. Gerstein, and M. Snyder, *RNA-Seq: a revolutionary tool for transcriptomics.* Nature reviews. Genetics, 2009. **10**(1): p. 57-63.
10. Kang, Y.-J., et al., *CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features.* Nucleic Acids Research, 2017. **45**(W1): p. W12-W16.
11. Guo, J.-C., et al., *CNIT: a fast and accurate web tool for identifying protein-coding and long non-coding transcripts based on intrinsic sequence composition.* Nucleic Acids Research, 2019. **47**(W1): p. W516-W522.
12. Li, A., 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,

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2014. **15**(1): p. 311.

13. Wang, L., et al., *CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model.* Nucleic acids research, 2013. **41**(6): p. e74-e74.
14. Wucher, V., et al., *FEELnc: a tool for long non-coding RNA annotation and its application to the dog transcriptome.* Nucleic Acids Research, 2017. **45**(8): p. e57-e57.
15. Camargo, A.P., et al., *RNAsamba: neural network-based assessment of the protein-coding potential of RNA sequences.* NAR Genomics and Bioinformatics, 2020. **2**(1).
16. Yang, C., et al., *LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning.* Bioinformatics, 2018. **34**(22): p. 3825-3834.
17. Fan, X.-N., et al., *lncRNA_Mdeep: An Alignment-Free Predictor for Distinguishing Long Non-Coding RNAs from Protein-Coding Transcripts by Multimodal Deep Learning.* International Journal of Molecular Sciences, 2020. **21**(15): p. 5222.
18. Yamashita, R., et al., *Convolutional neural networks: an overview and application in radiology.* Insights into Imaging, 2018. **9**(4): p. 611-629.
19. Kiranyaz, S., et al., *1D convolutional neural networks and applications: A survey.* Mechanical Systems and Signal Processing, 2021. **151**: p. 107398.
20. Acharya, U.R., et al., *Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network.* Information Sciences, 2017. **405**: p. 81-90.
21. Li, F., et al., *Feature extraction and classification of heart sound using 1D convolutional neural networks.* EURASIP Journal on Advances in Signal Processing, 2019. **2019**(1): p. 59.
22. Hsieh, C.-H., et al., *Detection of Atrial Fibrillation Using 1D Convolutional Neural Network.* Sensors (Basel, Switzerland), 2020. **20**(7): p. 2136.
23. Tjoa, E. and C. Guan, *A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI.* IEEE transactions on neural networks and learning systems, 2021. **32**(11): p. 4793-4813.
24. Ribeiro, M.T., S. Singh, and C. Guestrin, *"Why Should I Trust You?": Explaining the*

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Predictions of Any Classifier, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, Association for Computing Machinery: San Francisco, California, USA. p. 1135–1144.

25. Lundberg, S.M. and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions.* 2017: p. 4765--4774.
26. Chen, H., S. Lundberg, and S.-I. Lee, *Explaining Models by Propagating Shapley Values of Local Components*. 2019.
27. Shrikumar, A., P. Greenside, and A. Kundaje, *Learning important features through propagating activation differences*, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. 2017, JMLR.org: Sydney, NSW, Australia. p. 3145–3153.
28. Kong, L., et al., *CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine.* Nucleic acids research, 2007. **35**(Web Server issue): p. W345-W349.
29. Frankish, A., et al., *GENCODE reference annotation for the human and mouse genomes.* Nucleic Acids Research, 2018. **47**(D1): p. D766-D773.
30. Volders, P.-J., et al., *LNCipedia 5: towards a reference set of human long non-coding RNAs.* Nucleic Acids Research, 2018. **47**(D1): p. D135-D139.
31. Li, W. and A. Godzik, *Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.* Bioinformatics, 2006. **22**(13): p. 1658-1659.
32. Yates, A.D., et al., *Ensembl 2020.* Nucleic Acids Research, 2019. **48**(D1): p. D682-D688.
33. He, K., et al. *Identity Mappings in Deep Residual Networks*. 2016. Cham: Springer International Publishing.
34. Wang, S., et al., *Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model.* PLOS Computational Biology, 2017. **13**(1): p. e1005324.
35. Krogh, A., et al., *Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes.* J Mol Biol, 2001. **305**(3): p. 567-80.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

36. Mistry, J., et al., *Pfam: The protein families database in 2021.* Nucleic Acids Research, 2020. **49**(D1): p. D412-D419.

37. Blum, M., et al., *The InterPro protein families and domains database: 20 years on.* Nucleic Acids Research, 2020. **49**(D1): p. D344-D354.

38. Piovesan, D., et al., *MobiDB: intrinsically disordered proteins in 2021.* Nucleic Acids Res, 2021. **49**(D1): p. D361-d367.

39. Chen, X., et al., *Anti-proliferative and pro-apoptotic actions of a novel human and mouse ovarian tumor-associated gene OTAG-12: downregulation, alternative splicing and drug sensitization.* Oncogene, 2011. **30**(25): p. 2874-2887.

40. Anderson, Douglas M., et al., *A Micropeptide Encoded by a Putative Long Noncoding RNA Regulates Muscle Performance.* Cell, 2015. **160**(4): p. 595-606.

41. Makarewich, C.A., *The hidden world of membrane microproteins.* Experimental Cell Research, 2020. **388**(2): p. 111853.

42. Matsumoto, A. and K.I. Nakayama, *Hidden Peptides Encoded by Putative Noncoding RNAs.* Cell Structure and Function, 2018. **43**(1): p. 75-83.

43. Ulveling, D., et al., *Identification of a dinucleotide signature that discriminates coding from non-coding long RNAs.* Frontiers in genetics, 2014. **5**: p. 316-316.

44. Hartford, C.C.R. and A. Lal, *When Long Noncoding Becomes Protein Coding.* Molecular and Cellular Biology, 2020. **40**(6): p. e00528-19.

45. Choteau, S.A., et al., *MetamORF: a repository of unique short open reading frames identified by both experimental and computational approaches for gene and metagene analyses.* Database, 2021. **2021**.

46. Microsoft. *Neural Network Intelligence*. 2021; Available from: https://github.com/microsoft/nni.

47. Akiba, T., et al., *Optuna: A Next-generation Hyperparameter Optimization Framework*, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*. 2019, Association for Computing Machinery: Anchorage, AK, USA. p. 2623–2631.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Appendix

Table A1: Evaluation results of all tools on gorilla transcripts

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1-Score
Xlnc1DCNN	3,902	217	3,783	98	96.06	97.55	94.58	94.73	96.12
CPC2	3,878	281	3,719	122	94.96	96.95	92.98	93.24	95.06
CNIT	3,944	424	3,576	56	94.00	98.60	89.40	90.29	94.26
PLEK	3,847	685	3,315	153	89.53	96.18	82.88	84.89	90.18
CPAT	3,824	216	3,784	176	95.10	95.60	94.60	94.65	95.12
FEELnc	3,723	139	3,861	277	94.80	93.08	96.53	96.40	94.71
RNAsamba	3,899	214	3,786	101	96.06	97.48	94.65	94.80	96.12
lncRNA_Mdeep	3,863	217	3,783	137	95.58	96.58	94.58	94.68	95.62
LncADeep	3,842	158	3,842	158	96.05	96.05	96.05	96.05	96.05

Table A2: Evaluation results of all tools on chicken transcripts

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1-Score
Xlnc1DCNN	3,606	218	3,782	394	92.35	90.15	94.55	94.30	92.18
CPC2	3,679	198	3,802	321	93.51	91.98	95.05	94.89	93.41
CNIT	3,737	302	3,698	263	92.94	93.43	92.45	92.52	92.97
PLEK	3,119	756	3,244	881	79.54	77.98	81.10	80.49	79.21
CPAT	3,593	97	3,903	407	93.70	89.83	97.58	97.37	93.45
FEELnc	3,485	65	3,935	515	92.75	87.13	98.38	98.17	92.32
RNAsamba	3,618	100	3,900	382	93.98	90.45	97.50	97.31	93.75
lncRNA_Mdeep	3,538	131	3,869	462	92.59	88.45	96.73	96.43	92.27
LncADeep	3,586	109	3,891	414	93.46	89.65	97.28	97.05	93.20

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table A3: Evaluation results of all tools on mouse transcripts

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1
Xlnc1DCNN	15,307	1,680	14,320	693	92.58	95.67	89.50	90.11	92.81
CPC2	15,186	5,568	10,432	814	80.06	94.91	65.20	73.17	82.64
CNIT	15,530	3,473	12,527	470	87.68	97.06	78.29	81.72	88.74
PLEK	14,731	7,172	8,828	1,269	73.62	92.07	55.18	67.26	77.73
CPAT	14,812	2,186	13,814	1,188	89.46	92.58	86.34	87.14	89.78
FEELnc	14,244	1,281	14,719	1,756	90.51	89.03	91.99	91.75	90.37
RNAsamba	15,161	1,749	14,251	839	91.91	94.76	89.07	89.66	92.14
lncRNA_Mdeep	14,858	1,616	14,384	1,142	91.38	92.86	89.90	90.19	91.51
LncADeep	15,388	1,003	14,997	612	94.95	96.18	93.73	93.88	95.01

Table A4: Evaluation results of all tools on cow transcripts

Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1
Xlnc1DCNN	5,259	208	5,292	241	95.92	95.62	96.22	96.20	95.91
CPC2	5,201	308	5,192	299	94.48	94.56	94.40	94.41	94.49
CNIT	5,324	354	5,146	176	95.18	96.80	93.56	93.77	95.26
PLEK	4,751	767	4,733	749	86.22	86.38	86.05	86.10	86.24
CPAT	5,194	187	5,313	306	95.52	94.44	96.60	96.52	95.47
FEELnc	4,491	124	5,376	509	93.97	89.82	97.75	97.31	93.42
RNAsamba	5,294	191	5,309	206	96.39	96.25	96.53	96.52	96.39
lncRNA_Mdeep	5,188	169	5,331	312	95.63	94.33	96.93	96.85	95.57
LncADeep	5,262	121	5,379	238	96.74	95.67	97.80	97.75	96.70

Table A5: Top protein domains found within TPs compared with TNs

Protein Domain	Found in TPs	Found in TNs
Murine leukemia virus integrase, C-terminal	30	1
Domain of unknown function DUF1725	29	3
Reverse transcriptase domain	17	2
L1 transposable element, dsRBD-like domain	13	1
L1 transposable element, RRM domain	13	0
NADH:quinone oxidoreductase/Mrp antiporter, membrane subunit	13	0
Ribosomal protein S10 domain	9	5
Mos1 transposase, HTH domain	9	1
Retro-transcribing virus envelope glycoprotein	9	1
Integrase, catalytic core	8	1
DDE superfamily endonuclease domain	7	3
Ribosomal protein L23/L25, N-terminal	6	1
Cytochrome c-like domain	6	1
Ribosomal protein L30, ferredoxin-like fold domain	5	3
Domain of unknown function DUF4764	5	1
Reverse transcriptase/retrotransposon-derived protein, RNase H-like domain	5	0
Mitochondrial cytochrome c oxidase subunit VIc/VIIs	5	1
Cytochrome c oxidase subunit II-like C-terminal	5	0
Integrase, C-terminal, retroviral	5	3
Cytochrome b/b6, N-terminal	5	0

Table A6: Top protein families found within TPs compared with TNs

Protein Family	Found in TPs	Found in TNs
TLV/ENV coat polyprotein	23	2
Ribosomal protein L21e	18	1
High mobility group protein HMGN	16	5
BNIP3	11	3
Transposase, L1	11	0
Ribosomal protein L44e	10	2
Ribosomal protein S26e	10	0
Ribosomal protein S10	10	2
Ribosomal protein L34Ae	9	0
Ribosomal protein S27	7	2
Ribosomal protein S12e	7	0
Ribosomal protein S8	7	2
Vomeronasal receptor, type 1	7	0
Transposase, type 1	7	0
High mobility group protein HMGB1	7	0
FAM27D/FAM27E	6	0
Protein FAM27	6	0
NADH-ubiquinone reductase complex 1 MLRQ subunit	6	1
Elongin-C	6	4
Ribosomal protein S3Ae	6	2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

VITA

NAME	Rattaphon Lin
DATE OF BIRTH	4 April 1995
PLACE OF BIRTH	Bangkok

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University