CNCX : Classification of Coding and Noncoding Transcripts Based on eXtreme Gradient Boosting Classifier

by

Mahfida Jerin
Exam Roll: Curzon Hall-260
Registration No: 2014-916-629
Session: 2014-15

Mehreen Rahman
Exam Roll: Curzon Hall-266
Registration No: 2014-317-787
Session: 2014-15

A project submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF DHAKA

September 30, 2024

Declaration

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Dr. Sarker Tanveer Ahmed Rumee, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka. We also declare that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

.....

(Dr. Sarker Tanveer Ahmed Rumee)

Supervisor

Signature

.....

(Mahfida Jerin)

.....

(Mehreen Rahman)

Abstract

Both coding and noncoding RNA(ncRNA) transcripts play important roles in biological processes. Though ncRNA transcripts do not encode to protein, they act as a regulatory element in many such activities. Apart from that, ncRNA transcripts can also engage in causing many deadly human diseases like cancer, alzheimer etc. So, proper identification of ncRNA transcripts is important as finding the protein coding sites in genome. However, this task is not straight forward as they often exhibit similar positional and functional attributes like the coding RNA transcripts. Due to the abundance of ncRNAs, computational approaches have been widely adopted to distinguish between coding and noncoding transcripts. In this project, a new method (CNCX) has been proposed based on a new and improved machine learning technique, XGBoost, that provides a new direction and perspective to the feature selection process for the aforementioned classification problem. The proposed approach is validated through real life and popular data sets. Experimental results found that CNCX distinguishes coding and ncRNA transcripts with average accuracy rate 92.89%, which is at least 2 to 3 percent higher than the state of the art methods.

Acknowledgements

All praise is to the Almighty, who is the most gracious and most merciful. There is no power and no strength except with Him.

Our deepest gratitude goes to our thesis supervisor, Dr. Sarker Tanveer Ahmed Rumee, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka, for his proper guidance in our research field. He has shared his expert knowledge and has been an integral support in our thesis work by constantly keeping updates and urging us to something significant.

We want to thank our families and friends for their unwavering love and support. The opportunities that our parents have made possible for us, determines the personalities we have built and the work that we produce today.

Lastly, we want to thank the Department of Computer Science and Engineering, University of Dhaka, its faculty, staff, and all other individuals related to the department. The department has facilitated us throughout our undergraduate program and subsequent thesis, and has also formed the base for our future endeavours.

Mahfida Jerin
Mehreen Rahman
January, 2019

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

RNA transcripts can be divided into two sub-classes depending on their ability to be translated into protein : coding (mRNA) and noncoding transcript(ncRNA). Although, noncoding transcripts do not produce protein, they play an important role as a regulatory element in many biological processes, such as - RNA splicing, DNA replication, gene regulation, genome defense, hormonal balancing etc. Most importantly, ncRNA transcripts also engage in case of many deadly human diseases[7, 27] including Cancer[22] and Alzheimer[30]. For example, lncRNA PCA3 has been confirmed to be related to the formation of prostate cancer by showing 60 times expression levels in prostate tumors compared with normal tissues.

Apart from these, lots of existing studies have proved the significant role of ncRNAs in diverse biological processes in not just human, but also other species including plants. So, proper identification of noncoding transcripts have been critical to the knowledge of human biology and diseases. Among them, the long noncoding RNA transcripts (lncRNAs) are of particular interest. NcRNA transcripts with length of more than 200 nucleotides are referred as lncRNA transcript.

LncRNA transcripts often exhibit similar attributes as the coding transcripts but their functionalities are completely different. This makes their identification quite challenging.

1.1 Motivation

The study of lncRNA transcripts is vital to understand various biological processes. For example, researchers are constantly giving their effort to find the causes of various diseases. LncRNA transcripts are considered as one of the potential bio-marker that influences causing a number of diseases [27], such as lung cancer, breast cancer, colon cancer, prostate cancer, bladder cancer, thyroid cancer, ovarian cancer, diabetes, AIDS and many more. According to lncRNA Disease database (`http://www.cuilab.cn/lncrnadisease`), there have been more than 200 diseases related with various lncRNA transcripts and more than 300 lncRNA transcripts play crucial roles in various human complex diseases[2].

At the same time, next generation sequencing has provided researchers with millions of new sequences related with various diseases and human samples. Fast and efficient analysis of these data is critical to the timely discovery of disease factors and relevant drugs. As a result, computational approaches are widely being adopted to mine important information from these sequences, which include identifying lncRNA transcripts and distinguishing them from protein coding transcripts.

Among the various strategies to solve this problem, machine learning and deep learning algorithm based approaches are found to be more accurate and efficient. This work also adopts a machine learning based classifier to identify long noncoding RNA transcripts.

Although there are a large number of annotated lncRNA transcripts, only a few lncRNA transcripts have been extensively studied for the identification of their possible functionalities and underlying molecular mechanism[23]. Therefore, it is a big challenge for both experimental researches and computational biology to be combined to accurately identify the functionalities of lncRNA transcripts[10].

1.2 Problem Statement

In this project, the primary objective is to find an efficient method to distinguish between protein coding and noncoding RNA transcripts. Here, emphasis is given on selecting features which are biologically significant to differentiate these two types of transcripts.

Some noncoding transcripts behave like the protein coding ones and in some cases also located in the same areas of the genome. By analyzing the transcripts, it is also observed that these transcripts are structurally very similar to coding ones that go through the same transcription process[21]. This is why the noncoding transcripts are very difficult to distinguish from the coding transcripts.

So, to classify coding and noncoding transcripts, a machine learning based binary classifier is chosen to do this task, where the classifier model is trained with a novel feature set which is primarily composed of protein based features. The proposed classifier model is then able to identify lncRNA transcripts with high accuracy rate.

1.3 Contributions

The major contributions of this work are :

- This work proposes *CNCX*, a machine learning based classifier based on extreme gradient boosting (XGboost) classification algorithm. CNCX is capable of identifying coding and noncoding RNA transcripts from the DNA sequences.
- CNCX uses a novel set of sequence based features, which greatly influences the high accuracy of the proposed method.
- CNCX has been evaluated with real data sets (used by the state of the methods) and found to be more accurate in most of the cases. For these data sets, CNCX has been able to distinguish coding and noncoding RNA

transcripts with average accuracy rate 92.89%, which is at least 3 percent higher than the state of the art methods.

1.4 Organization of the Report

The report has been organized in a unique way. Some preliminary concepts and terminologies relevant to this work are introduced in **Chapter 2**. **Chapter 3** gives a summary of existing methods, especially machine learning based techniques to identify lncRNA transcripts. In **Chapter 4**, the proposed method is discussed in detail. The proposed algorithm is briefly compared to the referred existing algorithms in this chapter. The performance of the proposed technique is evaluated on the basis of the data sets used by the state of art methods and one independent data set. Details of this analysis can be found in **Chapter 5**. Finally, **Chapter 6** presents the summary contributions and the results obtained in this research along with some directions to possible future work.

Chapter 2

Background

This chapter introduces primary concepts and key terms which are necessary to understand the proposed method.

At first, a very basic introduction to the process of DNA to protein translation is given along with the definitions of coding and noncoding RNA. Then the details of the XGBoost classifier on which the proposed approach is based is discussed.

2.1 Central Dogma of Biology: DNA to Protein

Protein is produced due to the flow of genetic information from DNA to RNA. It is suggested by the central dogma that the information needed to produce all of the proteins are contained in DNA and that RNA acts as a messenger by carrying this information to the ribosomes. The ribosomes can be referred as factories in the cell as it is where the information is translated from a code into protein. The process by which the DNA instructions can be converted into protein is called gene expression. The process of converting DNA to RNA is shown in figure 2.1.

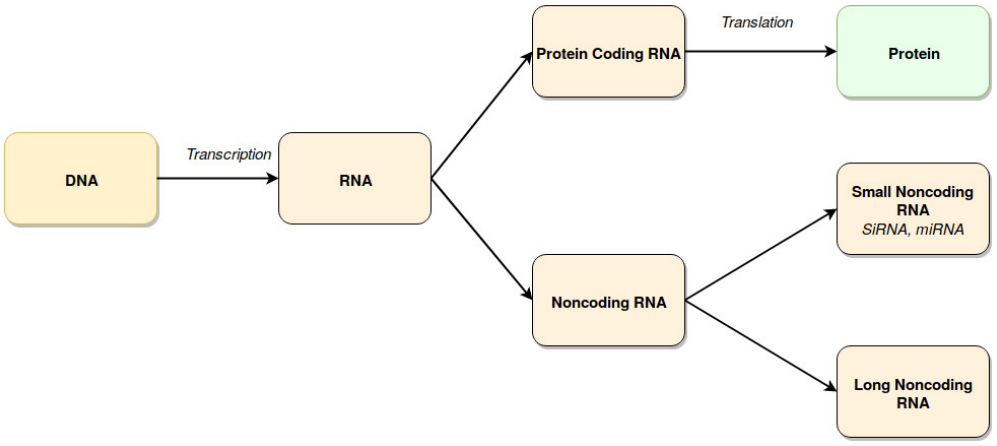

FIGURE 2.1: DNA to RNA conversion

The two key stages of gene expression are : transcription and translation. Transcription is the stage in which the DNA instruction of every cell is converted into small, portable RNA messages. During translation, these messages travel from the location of the DNA to the ribosomes where they are converted into specific proteins. The central dogma states that the patterns of information that occur most frequently in the cells are :

- From existing DNA to make new DNA, which happens in the DNA replication phase
- From DNA to make new RNA that happens in the transcription phase
- From RNA to make new proteins which takes place in the translation phase

2.2 Coding RNAs and Noncoding RNAs

Both coding and noncoding RNAs consist of nucleotide sequences. As stated previously, coding RNAs encode proteins necessary for cellular activities, whereas noncoding RNAs are unable to do so. This is the difference between these two types of RNA. Many sequential features play an important role in the identification

of coding and noncoding transcripts. Based on these features, scientists have been performing various experiments and even now, they are still focusing on discovering a faster and accurate method to distinguish between coding and noncoding RNAs.

The number of coding RNA is quite low compared to the number of noncoding RNA in the genome. In case of the human genome, percentages of coding and noncoding RNAs are about 2% and 98%.

Noncoding RNAs can be recognized biologically through experiments with full-length complementary DNA cloning and genomic tiling arrays in the transcriptomes of organisms. These technologies suit noncoding RNAs in an efficient way, but are costly as always require enough RNA samples. This is the limitation and to overcome this shortage, researchers felt the necessity to develop computational approaches [6] and incorporate these approaches in experimental methods[12].

2.3 XGBoost: eXtreme Gradient Boosting

XGBoost is designed in such a way that it can execute parallelism and also perform like a multicore computer and GPU. As a result, it is easier to train a large data set in an efficient way.

There are a number of attributes of protein coding and noncoding RNA transcripts that have made it important to identify the sequences correctly.

In this project, a new method is proposed to distinguish protein coding and noncoding RNA transcripts. The focus has been mainly on choosing features that are optimized and a model that is based on an advanced machine learning algorithm called XGBoost[5](eXtreme Gradient Boosting). This project has been conducted on a group of data sets containing coding and noncoding transcripts of the human genome. This method may not outperform other classifiers in many areas, but it has provided a new perspective.

2.3.1 Ensembling

Decision tree is a supervised learning method that is used for solving classification problems for its ability to analyze data that produces good results. Ensembling is a technique that can be used for better result and avoiding weak prediction in a single decision tree. This technique uses a number of decision trees and combines them to create a better prediction by reducing error. Two variations in ensembling are boosting and bagging. Bagging is a technique in which predictors are built independently, on the other hand, boosting is a technique in which predictors are built sequentially.

When trying to determine the label in case of classification problem, in most cases, some of the predicted labels differ from the target labels. The reasons are noise, variance and bias in the data. The solution to reduce these is ensembling.

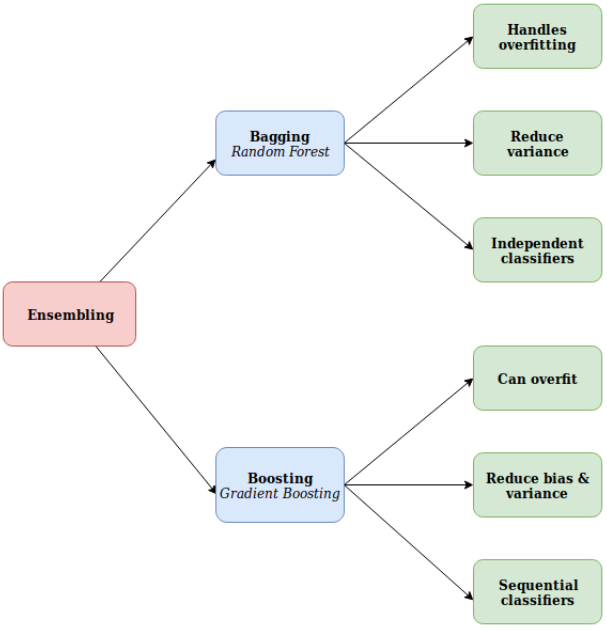

FIGURE 2.2: Ensembling

When a single predictor is used to predict the label, ensembling a number of different predictors will result in a better performance. The variations of using multiple predictors can be observed in both bagging and boosting as mentioned

previously. As the proposed method is based on an advanced version of boosting algorithm, boosting is discussed below :

2.3.2 Boosting

In case of boosting, the subsequent predictors learn from the mistakes of previous predictors. These predictors are basically decision trees, regressors, other types of classifiers etc. As current predictors learn from the mistakes of the previous ones, boosting takes lesser time to reach close to the final label.

A significant extension of boosting is gradient boosting. It uses gradient decent algorithm to minimize error. Several single decision trees with gradient decent algorithm are joined in a logical order[9]. Gradient boosting is used in regression, classification and ranking problem[19].

2.3.3 Gradient Boosting

Gradient boosting is a boosting algorithm. It is a machine learning technique that is used in classification and regression problem. In this technique, a final prediction model is produced by ensembling a number of weak prediction models, which are in most cases decision trees. So, the current model is fitted with residuals of the previous prediction and then the loss is minimized when adding the latest prediction.

The explanation behind the intuition of gradient boosting is to continuously reduce the patterns in residuals and building a strong model based on the weaker models to make the prediction better. Defining a loss function and minimizing it is the main goal of a supervised learning algorithm. If the mean squared error(MSE) is defined as loss:

$$Loss = MSE = \sum (y_{\mathrm{i}} - {y_{\mathrm{i}}}^{p})^2$$

where,

$$y_i = ith\ target\ value,$$

$$y_i{}^p = ith\ prediction,$$

$$L(y_i, y_i{}^p) is Loss function$$

The values where MSE is minimum can be found by using gradient descent and updating the predictions based on a learning rate.

$$y_i{}^p = y_i{}^p + a * \frac{\delta \sum (y_i - y_i{}^p)^2}{\delta y_i{}^p}$$

which becomes,

$$y_i{}^p = y_i{}^p - a * 2 \sum (y_i - y_i{}^p)^2$$

where,

$$a\ is\ learning\ rate\ and$$

$$\sum (y_i - y_i{}^p)^2\ is\ sum\ of\ residuals$$

So, the predictions need to be updated such that the sum of the residuals is close to 0 or minimum and predicted values are closer to the actual values.

An efficient way of gradient boosting is extreme gradient boosting which is known as XGBoost[5].

2.3.4 XGBoost Features

XGBoost is a highly flexible and versatile tool that can work through most regression, classification and ranking problems as well as user-built objective functions. XGBoost or eXtreme Gradient Boosting was developed by Tianqi Chen and now is part of a wider collection of open-source libraries developed by the Distributed Machine Learning Community (DMLC). It is a scalable and accurate implementation of gradient boosting machine. As it was constructed and developed only for the purpose of model performance and computational speed, it has proven to

push the limits of computing power for boosting tree algorithms. XGBoost offers a number of advanced features for model tuning, computing environments and algorithm enhancement. It has the capability to perform the three main forms of gradient boosting which are :

- Gradient Boosting (GB) algorithm also called gradient boosting machine including the learning rate.
- Stochastic GB with sub-sampling at the row, column and column per split levels.
- Regularized GB with both L1 and L2 regularization.

Also, it is robust enough to support fine tuning with the help of regularization of parameters. The system features that the XGBoost provides the user with are :

- Uses all of the CPU cores during training which is essential in obtaining parallelization of tree construction.
- Uses a cluster of machines that can assure distributed computing, which is essential for training very large models.
- Out-of-Core computing can be performed when data sets dont fit into memory.
- Can perform cache optimization of data structures and algorithm which ensures best use of hardware.

Some of the key algorithmic features that XGBoost provides are :

- Sparse aware implementation with automatic handling of missing data values.
- Block structure that supports the parallelization of tree construction.
- Continued training that ensures further boosting of an already fitted model on new data.

From the discussion above, it is observed that, XGBoost is based on the implementation of gradient boosting algorithm. Like the gradient boosting algorithm, this approach supports both regression, classification and predictive modelling problems. It ensures the best and the fastest performance that cannot be achieved by other implementations of gradient boosting.

As the structure of a noncoding RNA transcript is similar to a coding RNA transcript, it is difficult to find the noncoding transcripts which are equally important for the living beings as their coding counterparts. In chapter 3, several methods that are used to classify the coding and noncoding RNA transcripts are discussed. And the proposed algorithm to distinguish between coding and noncoding RNA transcripts using the extreme gradient boosting(XGBoost) is discussed in chapter 4.

Chapter 3

Related Work

Current biological databases contain a large amount of experimental data. With the continued research, the amount of biological data is growing exponentially. This exponential growth of data causes two problems : efficient storage and management of information and useful information extraction from the data.

The second problem is one of the main challenges in computational biology. To solve this problem, the development of tools and methods are required that are capable to transform heterogeneous data into biological knowledge required for the considered mechanism. The biological databases mainly contain nucleotides and amino acid sequences, biochemical networks, sequence data and protein and RNA structures. So the complexity of biological data ranges from simple strings to complex graphs; from 1-Dimensional to 3-Dimensional data. Considering the amount and complexity, it has become impossible for the experts to compute and compare entries within the databases.

Machine learning techniques, such as - Markov models, support vector machines(SVM), neural networks, decision trees etc. have been successful in analyzing biological data because of their capabilities to generalize and also handle randomness and uncertainty of data noise.

During the last few years, researchers have been heavily focused on using

various machine learning tools to distinguish between coding and noncoding transcripts. The proposed method is also a binary classifier taking help of automated machine intelligence.

Hence, this chapter limits its discussion of existing research closely relevant to the proposals that have been made in this project.

3.1 Support Vector Machine (SVM) Based Approaches

Coding-Potential Calculator (CPC)(2007)[17] evaluates the protein coding potential of transcripts using sequential features and support vector machine(SVM).The authors have used the 'libsvm' package to train SVM model with the standard radial basis function(RBF) kernel. These six sequential features are used to build the model : log-odds score, ORF coverage, ORF integrity, number of hits, hit score and frame score.

Coding Non-Coding Index(CNCI)(2013)[26] is developed for deploying sequence intrinsic composition to distinguish between protein coding and long noncoding(lnc) transcripts. This tool can also classify incomplete transcripts. In this tool, the classification is based on profiling adjoining nucleotide triplets (ANT). It effectively distinguishes between protein coding and noncoding sequences independent of known annotations. To train the data in SVM, the authors have used a library 'libsvm' using the standard radial basis function (RBF) kernel, where the C and gamma parameters have been set by default. These five features are extracted to build the model : MLCDS(Most-Like CDS), S-score of MLCDS, length-percentage, score-distance and codon-bias. The performance of this method is evaluated using the 10-fold cross-validation and ROC curve.

iSeeRNA(2013)[24] is developed with an SVM based classifier for the identification of long intergenic noncoding(linc) RNAs. 10 features in 3 categories(conservation, ORF and nucleotide sequences-based) have been used to build the model. The first

one is the conservation score, which is from the first category, ORF size and ORF coverage; these both are from the second category and the other seven features are from the third category which includes frequencies of seven di- or tri-nucleotide sequences (GC, CT, TAG, TGT, ACG and TCG). Performance of this model is evaluated using the 10 fold cross-validation.

Predictor of long non-coding RNAs and messenger RNAs based on an improved k-mer scheme(PLEK)(2014)[18] distinguishes long noncoding RNAs(lncRNAs) from messenger RNAs(mRNAs) in the absence of genomic sequences or annotations. This method uses improved k-mer scheme and SVM classifier to perform the prediction. 1,364 calibrated k-mer usage frequencies of each transcript have been regarded as computational features while performing the analysis. Performance is evaluated using the 10 fold cross-validation.

LncRScan-SVM(2015)[25] targets to distinguish long noncoding RNAs from the protein coding ones. using SVM. Features are derived from gene structure, transcript sequence, potential codon sequence and conservation. Six features are calculated : transcript length, standard deviation of stop codon counts between three translated frames, CDS score, exon count, exon length and average PhastCons scores of a transcript. Performance is evaluated based on the MCC(Matthews correlation coefficient) and AUC(area under the ROC curve).

CPC2 : a fast and accurate coding potential calculator based on sequence intrinsic features(2017)[15] this tool is nearly 1000 times faster than its first version, CPC[17]. Especially in the case of identifying lncRNAs, this tool performs very accurately. The core structure of the model is nearly same other than its feature selection and species neutrality. Like CPC, the model of this tool is constructed based on the SVM classifier. Four intrinsic features are used : Fickett TESTCODE score, ORF length, ORF integrity and isoelectric point (pI).

TABLE 3.1: Overview of Machine Learning Approaches

ML Approach	Method Name	Published Year	Trained Species	Input Format	Prediction Category
SVM	CPC	2007	multi-species	FASTA	pcRNA
	CNCI	2013	plant,human	FASTA,GTF	lncRNA
	iSeeRNA	2014	human	FASTA	lncRNA
	PLEAK	2014	human,plant	FASTA	lncRNA
	LncRscan-SVM	2015	human,mouse	GTF	lncRNA
	CPC2	2017	multi-species	FASTA,GTF	lncRNA
LR	CPAT	2013	human,mouse,fly,zebrafish	FASTA,BED	pcRNA
RF	LncRNA-ID	2015	human,mouse	BED,FASTA	lncRNA
	LncRNApred	2016	human	FASTA	lncRNA
DL	DeepLNC	2016	human	FASTA	lncRNA
	LncRNAnet	2018	human,mouse	FASTA	lncRNA
SVM,LR,RF,ELM.DL	LncFinder	2018	multi-species	FASTA	lncRNA

3.2 Logistic Regression (LR) Based Approach

Coding-Potential Assessment Tool(CPAT)(2013)[29] is an alignment-free method. It can distinguish coding and noncoding transcripts rapidly from a large set of candidates. A logistic regression(LR) model is used for the suitability of such binary classification. As it is an alignment-free method, all selected features (predictor variables) are calculated directly from the sequence. Four sequential features are used to build the model : ORF size, ORF coverage(the ratio of the ORF to

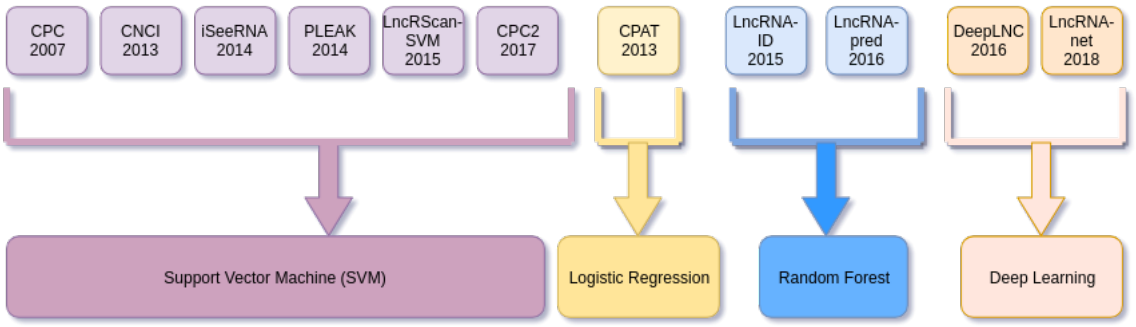

FIGURE 3.1: Overview of Different Machine Learning Methods to Find lncRNA

transcript length), Fickett TEST CODE score (termed as Fickett score now) and Hexamer usage bias.

3.3 Random Forest (RF) Based Approaches

LncRNA-ID(2015)[2] is a method to identify long noncoding(lnc) RNAs. Noncoding RNAs having a length of above 200 nucleotides are referred as lncRNAs. The model is constructed based on random forest (RF) classification. The features are derived from three different categories : ORF, ribosome interaction and protein conservation. 11 features are extracted : ORF length, ORF coverage, the next two are two Kozak motif-related features, the next three are ribosome coverage on three regions: transcript, ORF and 3'UTR(untranslated region), then ribosome release score, alignment score, alignment length with respect to profile HMM and the last one is alignment length with respect to the transcript. The performance is measured using AUC(the area under the ROC curve).

LncRNApred(2016)[20] can classify long noncoding(lnc) RNAs and protein coding transcripts fast and accurately. This tool is constructed based on the Random Forest Classifier(RFC). Three new features were introduced. The first one is MaxORF(the maximum length of ORF), the second one is RMaxORF(normalized MaxORF) and the last one is SNR(Signal to noise ratio). Though this tool was

trained on human coding and noncoding transcripts, it is claimed by the authors that it can also predict transcripts in case of other species. Performance is evaluated based on the AUC(area under the ROC curve).

3.4 Deep Learning (DL) Based Approaches

DeepLNC(2016)[28] is used to identify long noncoding(lnc) RNAs. This classifier is a Deep Neural Network(DNN) which is quite fast and efficient in the identification process compared to other existing classifiers. The only feature used for this classifier is the information content stored in k-mer pattern using manually annotated training data sets from LNCipedia and RefSeq database. This information content which is generated based on the Shannon entropy function is responsible for the improved classifier accuracy.

LncRNAnet: long non-coding RNA identification using deep learning (2018) [3] is used for identifying long noncoding(lnc) RNAs using deep learning approach. This method uses recurrent neural network (RNN) for sequencing RNA model . It also uses a convolutional neural network (CNN) for identifying stop codons to get an open reading frame (ORF) indicator. The algorithm consists of four steps : bucketing which is the process of arranging sequences according to their length, detecting ORF indicators for each sequence, encoding sequences and learning RNA sequences. The preprocessed sequence input data are trained for learning lncRNA sequences. LncRNAnet performed better than the other tools for short sequences. This method successfully learned features and showed 7.83%, 5.76%, 5.30% and 3.78% improvements over the alternatives on a human test set in terms of specificity, accuracy, F1-score and area under the curve(AUC).

3.5 Hybrid Approach

LncFinder: an integrated platform for long non-coding RNA identification using sequence intrinsic composition, structural information and physicochemical

property (2018) [11] uses five types of classifiers: support vector machine (SVM), random forest(RF), extreme learning machine (ELM), logistic regression(LR) and deep learning(DL) which is assessed by changing parameters. The final classifier is built in a way that produces the highest accuracy. The classification method consists of feature selection, classifier construction, evaluation and finally learning long noncoding(lnc) RNAs.

The features are heterologous and are obtained from three different sections : intrinsic composition of the sequence, multi-scale structural information and physicochemical property based on EIIP and fast Fourier transform (FFT). By researching on various species and classifiers, the authors have come to the conclusion that SVM yields the most accurate result. LncFinder is flexible and user-friendly. It can extract almost all classic alignment-free features proposed by other methods. LncFinder performs feature extraction, feature selection, classifier construction and performance evaluation easily and efficiently. The customization of features and use of machine learning algorithm effectively facilitates research on poorly explored species and lncRNA property analysis. The support of parallel computing greatly accelerates the feature selection process and classifier construction.

3.6 Other Relevant Work

From the above discussion, it can be seen that each of the methods has its own advantages and disadvantages. These methods use different machine learning algorithms and different set of features. It is seen that even when two methods use the same machine learning algorithm to construct their models, their results vary due to the selection of different features. Some methods use supervised and some uses unsupervised learning. So there are many factors that are essential for the correct identification of a transcript being coding or noncoding.

In this chapter, only machine learning approaches for classifying protein coding and noncoding RNAs are discussed as these approaches perform efficient computational analysis. But there are other methods that are based on sequence analysis[4], applying assembly methodology[14] and ribosome profiling[13].

In this work, a new algorithm has been proposed that uses a new and improved machine learning technique and provides a new direction to the classification approach. The details of the algorithm are described in chapter 4 and chapter 5, the performance of the proposed method is analyzed and the results are compared to some of the existing ones.

Chapter 4

The Proposed Approach

The preliminary concepts have been discussed in chapter 2 and chapter 3 discusses the existing methods to classify coding and noncoding RNA transcripts. This chapter gives detail description of the proposed method to distinguish ncRNA transcripts from the coding transcripts based on the intelligent feature selection and one of the latest machine learning algorithm: eXtreme Gradient Boosting(XGBoost) classifier.

4.1 Problem Description

As mentioned in previous chapter, the existing approaches use various machine learning algorithms and each approach uses different set of features to classify protein coding and noncoding transcripts.

Though various approaches have been implemented to find noncoding transcripts, researchers are still struggling to get a complete picture about it, due to various sequences and structure intrinsic features[16]. The proposed approach also uses a machine learning algorithm with a set of biologically significant feature set.

4.2 Approach Details

The high level steps of the proposed *CNCX* algorithm (input data processing, feature selection, classifier model construction and identifying inputs as codingnoncoding) are depicted in Figure. 4.1.

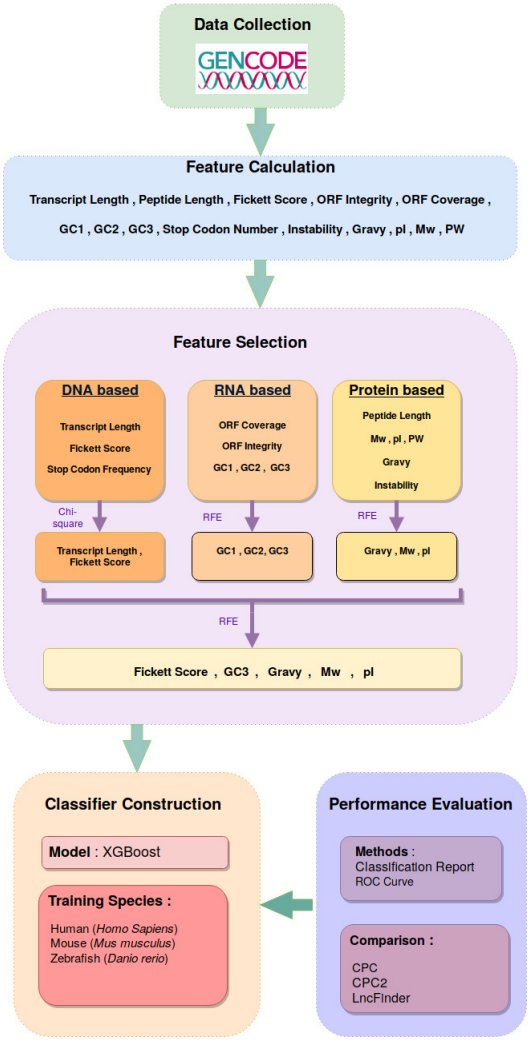

FIGURE 4.1: Flowchart of Proposed Approach

Based on the Figure.4.1, the basic work flow of the proposed approach is described below:

- **Data Collection** from various sources such as GENCODE[8], CPC2[15] and LncFinder[11].
- **Data processing** using a python script that converts the file containing FASTA sequences into a file containing corresponding categorical features.
- **Model construction** is done based on XGBoost classifier and validated on the training data with 10-fold cross validation.
- **Feature selection** using recursive feature elimination (RFE) on RNA-based and protein based features and chi-2 (χ^2) on DNA-based features and finally performing another RFE on combined selected features.
- **Tuning the parameters** of the predefined model based on the training data set and selected feature set that results in a tuned model
- Finally **Testing the model** with various data sets based on human, mouse and zebrafish species.

A flowchart for implementing the method CNCX is proposed in figure 4.1.

4.2.1 Data Collection

With the help of our supervisor, both training and testing data sets have been collected. Three training data sets have been collected from CPC2[15], lncFinder[11] and GENCODE[8]. First two of these data sets are the training data sets for the corresponding methods and the third data set is a combination of two data sets collected from GENCODE. The proposed approach has been tested on 4 different datatsets. One of these is collected from GENCODE[8] and the other 3 are the testing data sets of lncFinder[11].

4.2.2 Data Processing

A python script has been prepared that can calculate feature from FASTA sequences. All the collected data sets have been processed using this script that

produces files containing tabular data divided into 14 categorical features. These 14 features have been selected based on literature survey over 15+ peer-reviewed papers published recently. A flowchart for implementation of the proposed method CNCX is shown in figure 4.2.

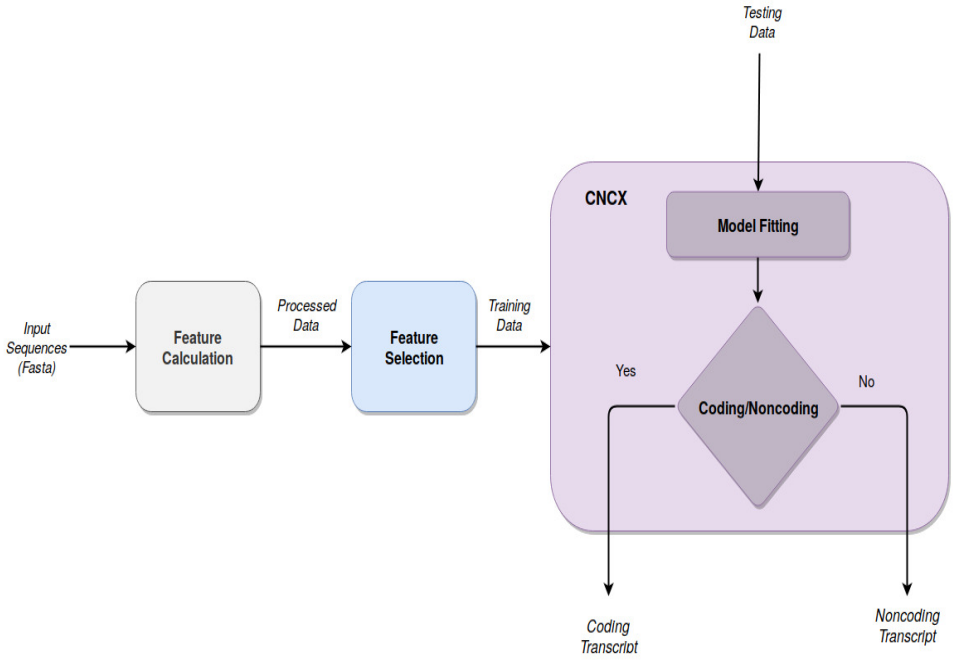

FIGURE 4.2: Working Process of the Proposed Approach

4.2.3 Feature Description

Below is a table 4.1 containing brief descriptions of the considered 14 features:

Table 4.1: Feature Description

Feature Name	**Description**
Transcript Length	The length of a single transcript
Peptide Length	The length of these amino acid polymeres.
Fickett Score	distinguishes protein coding RNAs and noncoding RNAs according to the combinational effect of nucleotide composition and codon usage bias. The Fickett score is independent of the ORF.
ORF Integrity	A boolean that determines whether an ORF starts with a start codon and ends with a stop codon.Start codon is compared to "ATG" and stop codons are compared to "TAG" ,"TAA" and "TGA".
ORF Coverage	The ratio of ORF length and transcript length. $ORF_{\text{coverage}} = \frac{ORF_{\text{length}}}{transcript_length}$
GC1	GC content in the first position of codons.
GC2	GC content in the second position of codons.
GC3	GC content in the third position of codons.
Stop Codon Number	The number of STOP Codons in the transcript.
Instability	An estimation of the stability of a predicted peptide.
Gravy	Grand average of hydropathicity of a predicted peptide.
Isoelectric Point	Theoretical isoelectric point of a predicted peptide.
Mw	Molecular weight of a predicted peptide.
pI	Theoretical isoelectric point of a predicted peptide.
PW	The ratio of pI and Mw. $PW = \frac{pI}{Mw}$

4.2.4 Feature Selection

The selected features (14 in total) can be categorized into 3 types : protein based, DNA based and RNA based. In total, 2 types of feature selection, Recursive Feature Elimination(RFE) and Chi-square Feature Selection(Chi2), have been performed four times.

RFE fits a model and removes the weakest feature or features until the specified number of features is reached. In chi2, chi-square statistics are calculated

between every feature variable and the target variable and the relationship between these two is observed. If the target variable is independent of the feature variable, the feature variable is discarded. Otherwise, the feature variable is considered very important.

Feature selection process is described in detail in Algorithm 1.

Algorithm 1 Feature Selection

```
procedure FEATURE SELECTION(trainingdata)
    model ← XGBClassifier()
    Features ← [DNA based feature,RNA based feature,protein based feature]

    procedure DNA BASED FEATURE SELECTION(x, y) :
        DNA ← chi2(x, y)
        k ← chi DNA base feature(No.of features ← 2)
        for i ← 0 to len of k do
            if k[i] ← True then
                final feature ← dna based feature[i]
    procedure RNA BASED FEATURE SELECTION(x, y) :
        RNA ← RFE(x, y)
        k ← rfe RNA base feature(No.of features ← 3)
        for i ← 0 to len of k do
            if k[i] ← True then
                final feature ← rna based feature[i]
    procedure PROTEIN BASED FEATURE SELECTION(x, y) :
        Protein ← rfe(x, y)
        k ← rfe protein base feature(No.of features ← 3)
        for i ← 0 to len of k do
            if k[i] ← True then
                final feature ← protein based feature[i]
    procedure FINAL FEATURE SELECTION(x, y) :
        Total ← rfe(x, y)
        k ← rfefinalfeature(No.of features ← 5)
        for i ← 0 to len of k do
            if k[i] ← True then
                final selected total feature ← final feature[i]
    return selected total feature
```

To perform this feature selection, firstly, the data set is prepared joining the three individual training sets. Then, an XGBoost model is instantiated having a particular set of parameters fixed which are often considered default[1]. In this

four staged feature selection approach, RFE is performed on the protein based features that selects the best 3 out of the 6 features. It happens in the procedure protein based feature selection in 1 In this case, pI, Mw and gravy are selected. Then again RFE is performed on the RNA based features which takes place in the procedure RNA based feature selection in 1 and 3 out of the total 5 features are selected. The selected features are : GC1, GC2 and GC3. Lastly, on the DNA based features, chi2 is performed that selects 2 out of the 3 features which are transcript length and stop codon number and it happens in the procedure DNA based feature selection in 1. These 8 features are selected in total on which the last RFE is performed and the best 4 features are selected. This takes place in the procedure final feature selection in 1. These features are : gravy, Mw, pI and GC3. It can be observed that none of the RNA based features are selected. So, an RNA based feature is added manually and this feature has been selected based on prior literature survey knowledge. This feature is Fickett score which has proved to perform well in the case of the experiments performed in the past approaches.

Among the 4 features selected, 3 of them them are protein based and the other one is DNA based. Using these 5 features and particular training sets, the parameters of the model are tuned.

4.2.5 Model Construction and Parameter Optimization

This work uses the state of the art XGBoost algorithm for its base classifier model. However, to use it solve the problem in hand, detailed parameter tuning have been performed. The details of parameter are included in the discussion below.

XGBoost has 3 types of parameters that need to be set. General, booster and learning task parameters. General parameters are tuned to guide the overall functioning, booster parameters guide the individual tree or regressor at each step and learning task parameters are tuned to guide the optimization performed.

In this approach, Gamma, min_child_weight, max_depth, subsample, colsample_bytree and learning_rate are the parameters which are tuned. As tree booster

always outperforms linear boosters, these booster parameters are most frequently tuned. Also, the general approach is to tune tree based parameters with lower learning rates. So, in this approach, the focus is to tune these parameters only.

The initial values that are set for the parameters are : $learning\_rate = 0.1$, $n\_estimators = 1000$, $max\_depth = 5$, $min\_child\_weight = 1$, $gamma = 0$, $subsample = 0.8$, $colsample\_bytree = 0.8$, $objective =' binary : logistic'$, $nthread = 4$, $scale\_pos\_weight = 1$ and $seed = 27$. These initial estimates will be tuned anyways.

After this initialization, max_depth and min_child_weight is tuned. In this approach, tuning is based on grid search,

The goal of parameter optimization is to find the set of values of the parameters for which validation error function minimizes. This validation error function is very expensive to evaluate. All possible combinations of the parameters need to be considered and tested in the model which is very impractical. So a bunch of values for each of the parameters are selected against which the validation error function is calculated. The set of parameters for which the validation error is minimized most, it is chosen. When these sets of parameters are plotted in space, looks like a grid. Which is why it is referred as grid search.

In this approach, for three training data sets, this optimization is performed thrice. In all cases, max_depth and min_child_weight is tuned first. Grid search have been performed twice to get the optimal values for both the parameters. Using these two, gamma has been optimized next. One grid search has been enough to get the optimal gamma. Using these 3 parameters, subsample and colsample_bytree are tuned. To get the optimal values for this pair, grid search has been performed twice. Then using all these values, learning rate has been optimized and one grid search has been enough in this case.

After getting the optimized sets of parameters, the models are fitted based on the corresponding training sets.

4.2.6 Training and Testing the Model

For each training set, after the hyper parameters have been optimized, the model is instantiated accordingly. Then, the model is fit based on the 10-fold cross validation. After the model has been trained, it is tested on particular testing data sets and various performance metrics are calculated such as accuracy, recall, precision, f1-measure and roc-score.

4.3 Conclusion

The proposed *CNCX* algorithm provides the complete details on how RNA transcripts can be classified into coding and noncoding in an efficient way. It will classify the FASTA sequences in a data set faster than some of the existing methods for classifying transcripts and accurately as well. It will also consume less memory as the machine learning concept on which the classifier is based on is XGBoost which is designed to be memory efficient. This can be applied in real life data sets based on which the performance evaluation is observed in chapter 5.

Chapter 5

Results

In this chapter, the overall performance evaluation of the proposed approach(CNCX) is presented. For evaluation purposes, several real life data sets were used and the findings are compared to the state of the art methods.

5.1 State of the Art Methods and Data Sets

Among the recent tools, CPC, CPC2 have great significance. CPC2 is the successor of CPC. Both of these approaches use the same training set, which is actually the training set of CPC another prominent method to find lncRNAs in the wild.

Both CPC and CPC2 have been tested for the considered testing sets and the results are analyzed.

Results are also compared with LncFinder, the most recent and up to date method. LncFinder is capable of revealing the properties of lncRNA and mRNA from various perspectives and also capable of inspiring lncRNA-protein interaction prediction and lncRNA evolution analysis. It has been observed that the LncFinder has the ability to classify the transcripts more accurately then many of its predecessors.

The performance evaluation of CNCX has been determined by training it with the training set of CPC2 and LnFinder and the third time training it with

the considered training data set that has been prepared by combining 4 different data sets collected from GENCODE. Then the testing is performed twice for the two different training sets on 3 different data sets of 3 different species which are human, mouse and zebrafish (collected from GENCODE[8]).

5.2 Environment Setup for Experiment

The project is built and run on Core i5 processor based desktop PC running Ubuntu 18.04 OS. The system RAM was 8 GB at experiment time.

All source codes are written in python. Jupyter notebook has been used as the IDE. Additional packages which had to be added to the environment are listed below:

- biopython version-1.72
- python version 2.7
- XGBoost version-0.81
- scikit-learn 0.20.1
- numpy 1.15.4
- pandas 0.23.4
- pip 18.1

5.3 Performance Metrics

Classification Report shows a representation of the main classification metrics which are precision, recall, F1-score and support scores on a per-class basis for model.

The performance evaluation of CNCX is based on the following well known metrics:

$$Accuracy = \frac{TP + TN}{P + N}$$

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

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

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

$$F1 - score = \frac{2 \cdot Sensitivity \cdot PPV}{Sensitivity + PPV}$$

ROC Curve is another performance measure that shows the trade off between sensitivity and specificity meaning any increase in sensitivity will be accompanied by a decrease in specificity. The closer the curve follows the left-hand border and then the top border of the ROC space, The test will be more accurate. If the curve comes closer to the 45-degree diagonal of the ROC space, the test becomes less accurate.

5.4 Data Sets

5.4.1 Training Data Sets

Both the training and testing data sets considered contain FASTA sequences.

The first training data set is collected from the LncFinder[11] and is actually a combination of human and mouse sequences containing about 24,400 transcripts. Among them 16,000 are human and the rest are of mouse. This combined data set has a 50-50 distribution of coding and noncoding transcripts. This data set is referred to as Training Set 1 in the following discussions.

The second data set is the training sample used in CPC2[15], which contains about 28,436 transcripts: 17,984 coding and 10,452 noncoding.This data set is referred to as Training Set 2 in the following discussions.

The third training data set is a traininng sample created with latest GENCODE release[8] GENCODE version 29 human and GENCODE version 19 mouse, with contain about 15,000 coding and 15,000 noncoding human 'fasta' sequences along with 14998 coding and 14998 noncoding mouse species transcripts.

The full description of training data sets and its sources is shown in Table 5.1.

TABLE 5.1: Test Data Description

Training Data Set	**Database Source**	**Description**
Training Data Set 1	GENCODE (1),Ensembl (2)	Protein Coding Transcripts : 12200 noncoding Transcripts : 12000
Training Data Set 2	CPC2 (3)	Protein Coding Transcripts: 17984 noncoding Transcripts: 10452
Training Data Set 3	GENCODE (4,5)	Protein Coding Transcripts: 29998 noncoding Transcripts: 29998

[1] Human and Mouse sequences are collected from LncFinder which are gathered from GENCODE

[2] Zebrafish sequences are collected from LncFinder which are gathered from Ensembl

[3] Human, mouse, zebrafish, fly, worm and the model plant Arabidopsis sequences are collected from CPC2 which are gathered from RefSeq database, Swiss-Prot and Ensembl

[4].GENCODE Version 29 human coding and noncoding fasta sequences

[5].GENCODE Version 19 mouse coding and noncoding fasta sequences

5.4.2 Test Data Sets

The data set collected from GENCODE[8] is a combination of 56934 transcripts among which 50% are coding and the rest are noncoding. The other 3 data

sets collected from LncFinder[11] are of 3 different species. The human data set contains 5000 transcripts among which 50% are coding and the rest is noncoding. The mouse data set contains 3600 transcripts of which the half are coding and the rest are noncoding. The zebrafish data set contains about 7,982 transcripts which has a 50-50 distribution of coding transcripts and noncoding transcripts. The sources of the testing data is given in table 5.2.

TABLE 5.2: Test Data Description

Testing Data set	**Database**	**Species**	**Description**
Testing Data Set 1	GENCODE	Homo sapiens	Protein Coding Transcripts : 2500 noncoding Transcripts : 2500
Testing Data Set 2	GENCODE	Mus musculus	Protein Coding Transcripts: 1800 noncoding Transcripts: 1800
Testing Data Set 3	Ensembl	Danio rerio	Protein Coding Transcripts: 3991 noncoding Transcripts: 3991
Testing Data Set 4	GENCODE	Homo sapiens	Protein Coding Transcripts: 28467 noncoding Transcripts: 28467
Testing Data Set 5	GENCODE	Mus musculus	Protein Coding Transcripts: 18065 noncoding Transcripts: 18065

1. *Human fasta sequences are collected from testing samples of LncFinder which are gathered from GENCODE*

2. *Mouse fasta sequences are collected from testing samples of LncFinder which are gathered from Ensembl*

3. *Zebrafish sequences are collected from testing samples of LncFinder which are gathered from Ensembl*

4. *GENCODE Version 28 human coding and noncoding fasta sequences*

5. *GENCODE Version 18 mouse coding and noncoding fasta sequences*

5.5 Evaluation Result on Training Set 1

5.5.1 Test Data Set : Human (*Homo Sapiens*)

The *Accuracy* of LncFinder and CNCX are 96.76% and **96.22%**.

For human , the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.3 :

TABLE 5.3: Comparison between CNCX and LncFinder on Test Data Human

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.970	**0.959**	0.965	**0.965**
Recall/Sensitivity	0.965	**0.966**	0.970	**0.959**
F1-score	0.968	**0.962**	0.968	**0.962**
Number of Transcripts	2500		2499	

For human, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.1 :

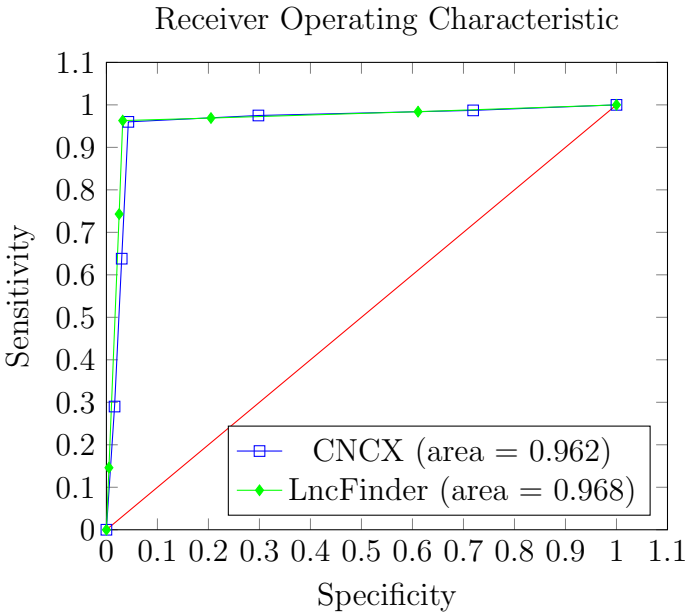

FIGURE 5.1: LncFinder and CNCX Comparison ROC Curve on Human Test Data Set on Training Data Set 1

5.5.2 Test Data Set : Mouse (*Mus musculus*)

The *Accuracy* of LncFinder and CNCX are 92.28% and **90.53%** .

For mouse, the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.4 :

TABLE 5.4: Comparison between CNCX and LncFinder on Test Data Mouse

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.904	**0.877**	0.943	**0.938**
Recall/Sensitivity	0.946	**0.942**	0.899	**0.968**
F1-score	0.925	**0.909**	0.921	**0.902**
Number of Transcripts	1800		1800	

For mouse, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.2 :

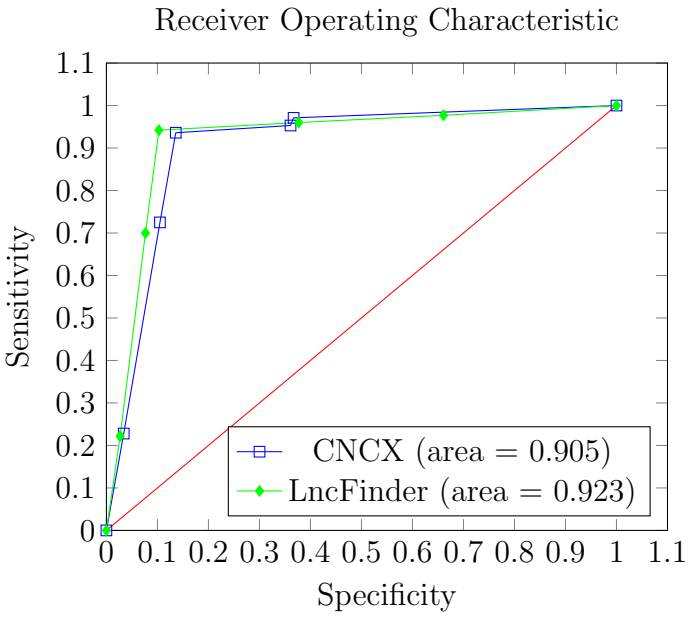

FIGURE 5.2: LncFinder and CNCX Comparison ROC Curve on Mouse Test Data Set on Training Set 1

5.5.3 Test Data Set : Zebrafish (*Danio rerio*)

The *Accuracy* of LncFinder and CNCX are 91.80% and **87.33%** .

For zebrafish, the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.5 :

TABLE 5.5: Comparison between CNCX and LncFinder on Test Data Zebrafish

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.907	**0.866**	0.930	**0.881**
Recall/Sensitivity	0.932	**0.884**	0.904	**0.862**
F1-score	0.919	**0.875**	0.917	**0.872**
Number of Transcripts	3991		3991	

For zebrafish, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.3 :

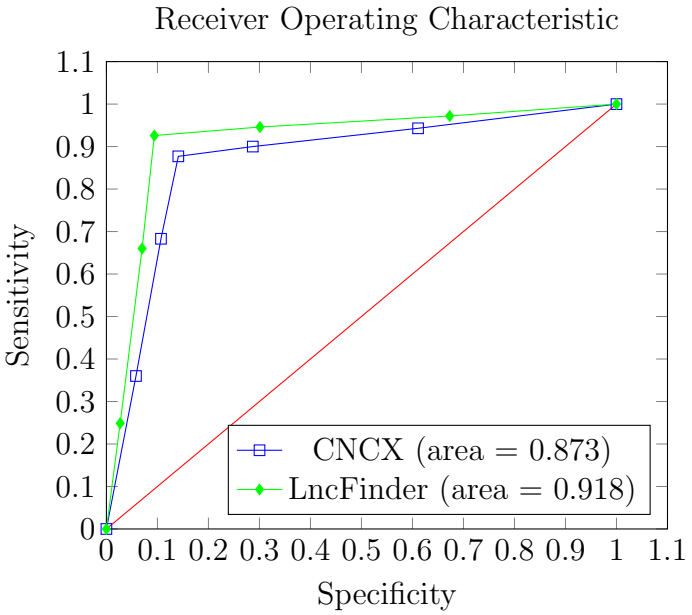

FIGURE 5.3: LncFinder and CNCX Comparison ROC Curve on Zebrafish Test Data Set on Training Data Set 1

5.5.4 Test Data Set : GENCODE version 28 Human

The *Accuracy* of LncFinder and CNCX are 89.34% and **87.73%** .

For GENCODE v28 human, the comparison between **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.6 :

TABLE 5.6: Comparison of CNCX and LncFinder on GENCODE v28 Human

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.946	**0.825**	0.955	**0.950**
Recall/Sensitivity	0.961	**0.958**	0.826	**0.797**
F1-score	0.900	**0.886**	0.886	**0.866**
Number of Transcripts	28467		28467	

For GENCODEversion 28 huuman, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.4 :

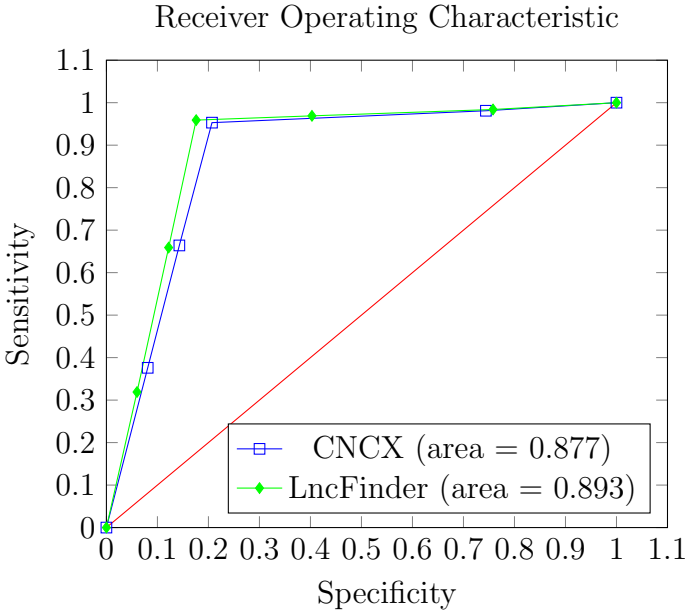

FIGURE 5.4: LncFinder and CNCX Comparison ROC Curve on GENCODE v28 Human Species on Training Data Set 1

5.5.5 Test Data Set : GENCODE version 18 Mouse

The *Accuracy* of LncFinder and CNCX are 92.22% and **89.14%** .

For GENCODE v18 mouse, the comparison between **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.7 .

TABLE 5.7: Comparison of CNCX and LncFinder on GENCODE v18 Mouse

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.908	**0.847**	0.937	**0.948**
Recall/Sensitivity	0.939	**0.955**	0.905	**0.828**
F1-score	0.923	**0.898**	0.921	**0.884**
Number of Transcripts	17855		17855	

For GENCODE v18 mouse, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.5 :

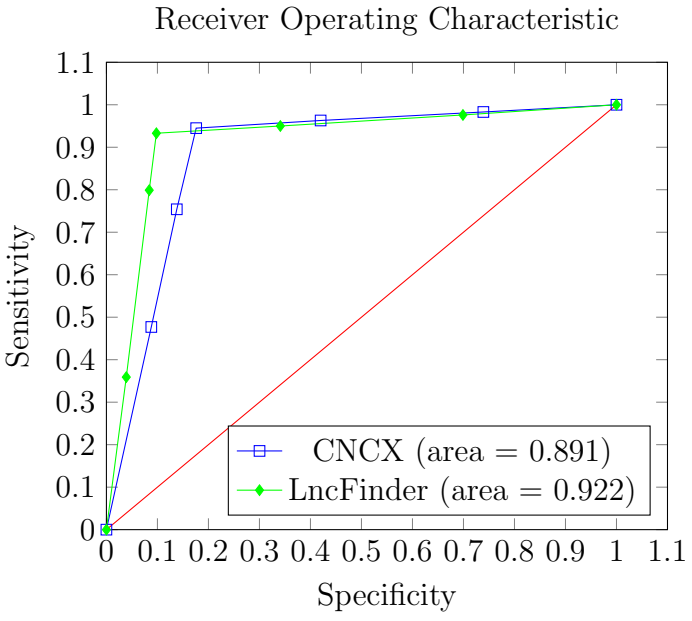

FIGURE 5.5: LncFinder and CNCX Comparison ROC Curve on GENCODE v18 Mouse Species on Training Data Set 1

5.6 Evaluation Result on Training Set 2

5.6.1 Test Data Set : Human (*Homo Sapiens*)

The *Accuracy* of CPC, CPC2, LncFinder and CNCX are 91.64%, 96.14%, 96.38% and **97.00%**.

For human data set, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** based on the **classification report** can be observed from table 5.8 :

TABLE 5.8: Comparison of CPC, CPC2, LncFinder and CNCX on Human

Criteria	**Noncoding**				**Coding**			
	CPC	**CPC2**	**LncFinader**	**CNCX**	**CPC**	**CPC2**	**LncFinader**	**CNCX**
Precision	0.985	0.953	0.975	**0.979**	0.865	0.970	0.953	**0.962**
Recall/Sensitivity	0.846	0.971	0.952	**0.961**	0.987	0.952	0.976	**0.979**
F1-score	0.910	0.962	0.963	**0.970**	0.922	0.961	0.964	**0.970**
Number of Transcripts	5000				5000			

For human, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.6 :

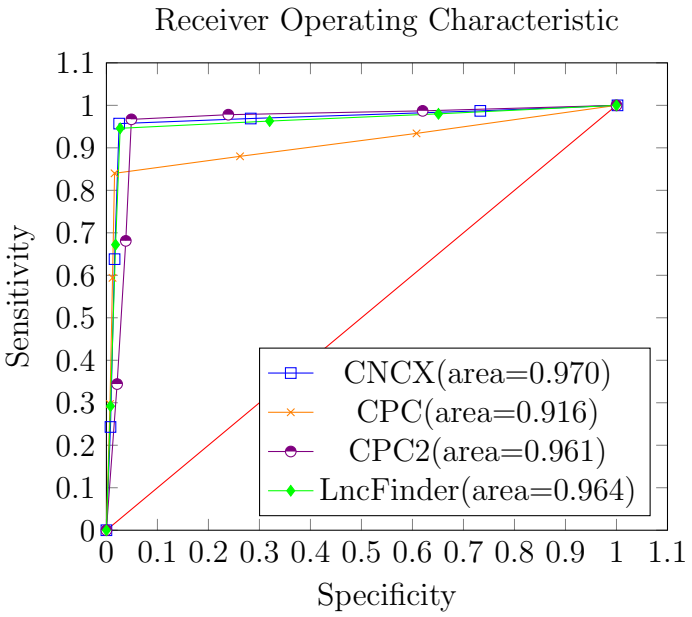

FIGURE 5.6: CPC, CPC2, LncFinder, CNCX Comparison ROC Curve on Human Data Set on Training Data Set 2

5.6.2 Test Data Set : Mouse (*Mus musculus*)

The *Accuracy* of CPC, CPC2, LncFinder and CNCX are 85.92%, 86.11%, 91.31% and **90.22%**.

For mouse, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** based on the **classification report** can be observed from table 5.9 :

TABLE 5.9: Comparison of CPC, CPC2 , LncFinder and CNCX on Test Data Mouse

Criteria	**Noncoding**				**Coding**			
	CPC	**CPC2**	**LncFinader**	**CNCX**	**CPC**	**CPC2**	**LncFinader**	**CNCX**
Precision	0.980	0.918	0.905	**0.898**	0.787	0.818	0.922	**0.906**
Recall/Sensitivity	0.733	0.793	0.923	**0.907**	0.985	0.929	0.903	**0.897**
F1-score	0.839	0.851	0.914	**0.903**	0.875	0.870	0.912	**0.902**
Number of Transcripts	3600				3600			

For mouse, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.7 :

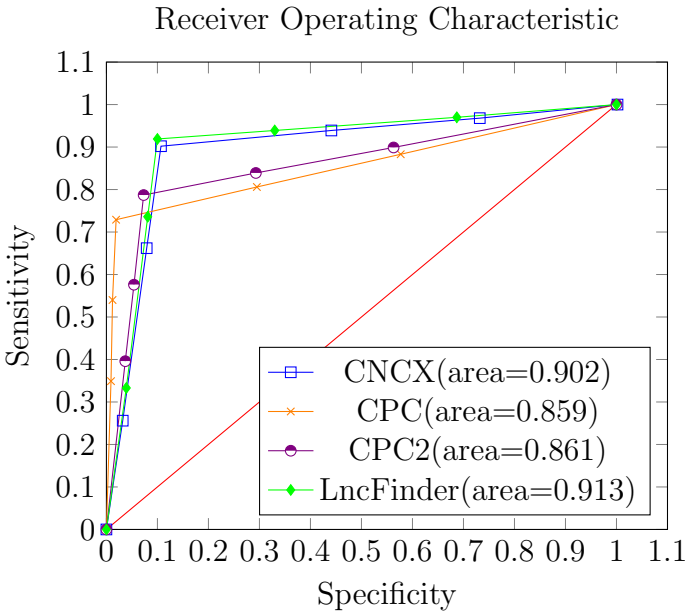

FIGURE 5.7: CPC, CPC2, LncFinder, CNCX Comparison ROC Curve on Mouse Data Set on Training Data Set 2

5.6.3 Test Data Set : Zebrafish (*Danio rerio*)

The *Accuracy* of CPC2, LncFinder and CNCX are 83.91%, 87.41% and **86.59%**.

For zebrafish, the comparison among **CNCX**, **CPC2** and **LncFinder** based on the **classification report** can be observed from table 5.10 :

TABLE 5.10: Comparison of CPC2, LncFinder and CNCX on Test Data Zebrafish

Criteria	**Noncoding**			**Coding**		
	CPC2	**LncFinader**	**CNCX**	**CPC2**	**LncFinader**	**CNCX**
Precision	0.882	0.893	**0.881**	0.805	0.897	**0.852**
Recall/Sensitivity	0.783	0.851	**0.846**	0.895	0.898	**0.886**
F1-score	0.830	0.871	**0.863**	0.848	0.877	**0.869**
Number of Transcripts	7982			7982		

For zebrafish, the comparison among **CNCX**, **CPC2** and **LncFinder** based on the **ROC charcateristics** can be observed from graph 5.8 :

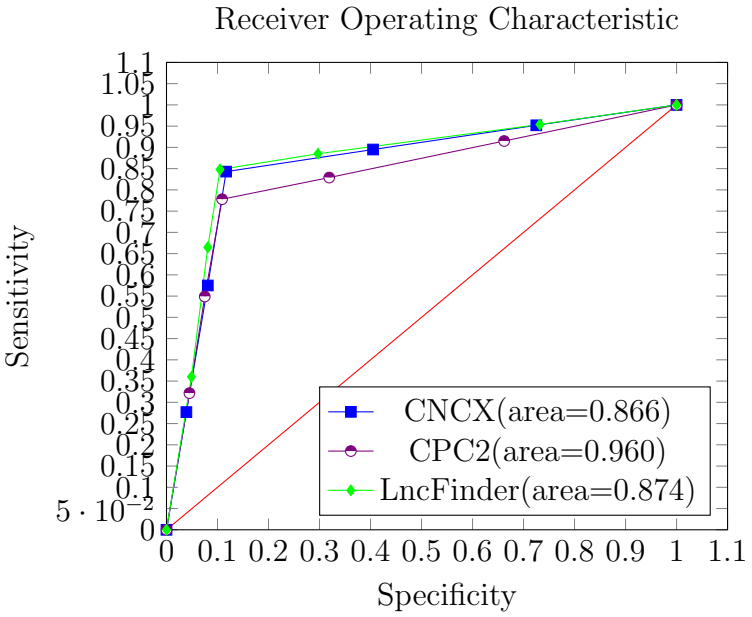

FIGURE 5.8: CPC2, LncFinder, CNCX Comparison ROC Curve on Zebrafish Species on Training Data Set 2

5.6.4 Test Data Set : GENCODE version 28 Human

The *Accuracy* of CPC, CPC2, LncFinder and CNCX are 81.71%, 81.86%, 85.13% and **89.04%**.

Based on the classification report, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** for GENCODE v28 human can be observed from table 5.11 :

TABLE 5.11: Comparison of CPC, CPC2, LncFinder, CNCX on GENCODE v28 Human

Criteria	**Coding**				**Nonoding**			
	CPC	**CPC2**	**LncFinader**	**CNCX**	**CPC**	**CPC2**	**LncFinader**	**CNCX**
Precision	0.986	0.924	0.844	**0.855**	0.735	0.755	0.858	**0.934**
Recall/Sensitivity	0.643	0.694	0.861	**0.941**	0.991	0.943	0.841	**0.840**
F1-score	0.779	0.93	0.853	**0.896**	0.844	0.839	0.850	**0.885**
Number of Transcripts	56934				56934			

For GENCODE v28 human, comparison among CPC, CPC2, LncFinder and CNCX can be observed from the graph 5.9 :

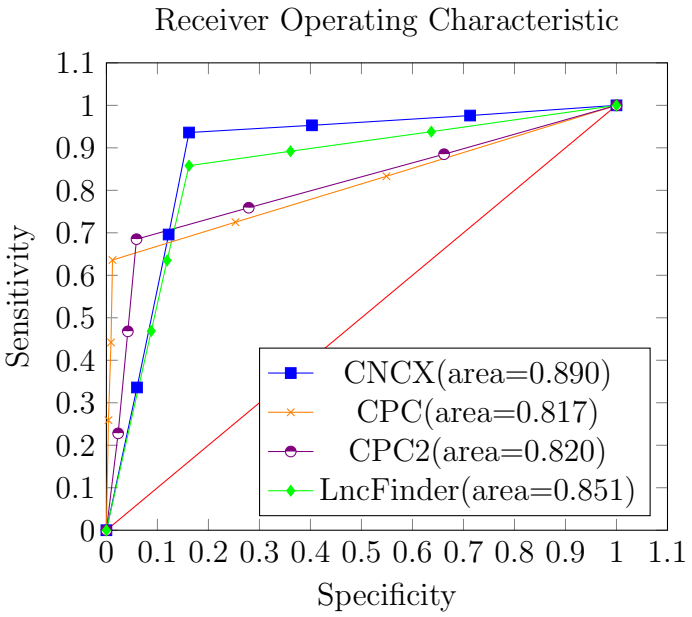

FIGURE 5.9: CPC, CPC2, LncFinder, CNCX Comparison ROC Curve on GENCODE v28 Human Species on Training Set 2

5.6.5 Test Data Set : GENCODE version 18 Mouse

The *Accuracy* of CPC, CPC2, LncFinder and CNCX are 81.71%, 81.86%, 92.22% and **89.18%**.

Based on the classification report, the comparison among **CNCX**, **CPC**, **CPC2** and **LncFinder** for GENCODE v18 mouse can be observed from table 5.12 :

TABLE 5.12: Comparison of CPC, CPC2, LncFinder and CNCX on GENCODE v18 Mouse

Criteria	**Coding**				**Noncoding**			
	CPC	**CPC2**	**LncFinader**	**CNCX**	**CPC**	**CPC2**	**LncFinader**	**CNCX**
Precision	0.981	0.927	0.908	**0.871**	0.761	0.791	0.937	**0.915**
Recall/Sensitivity	0.690	0.752	0.939	**0.919**	0.987	0.941	0.905	**0.864**
F1-score	0.810	0.830	0.923	**0.895**	0.859	0.859	0.921	**0.889**
Number of Transcripts	56934				56934			

For GENCODE v18 mouse, comparison among CPC, CPC2, LncFinder and CNCX can be observed from the graph 5.10 :

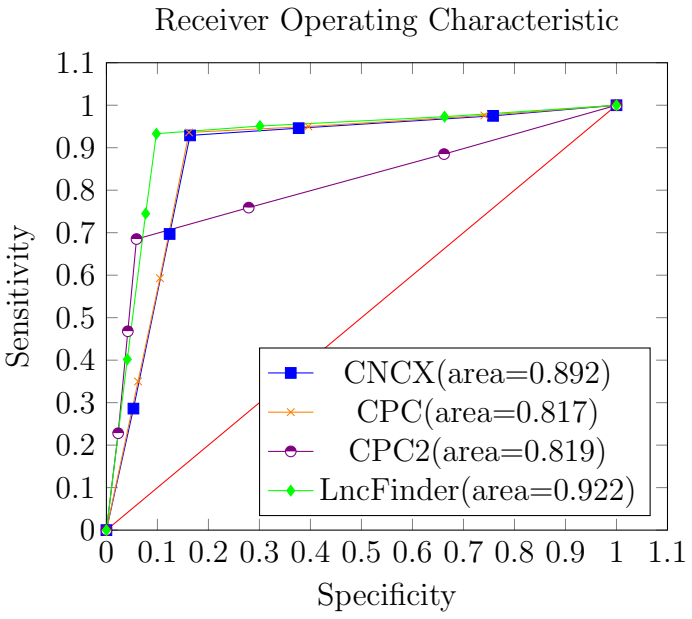

FIGURE 5.10: CPC, CPC2, LncFinder, CNCX Comparison ROC Curve on GENCODE v18 Mouse Species on Training Set 2

5.7 Evaluation Result on Training Set 3

5.7.1 Test Data Set : Human (*Homo Sapiens*)

The *Accuracy* of LncFinder and CNCX are 96.02% and **95.80%**.

For human , the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.13 :

Table 5.13: Comparison between CNCX and LncFinder on human Test Data

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.973	**0.965**	0.948	**0.951**
Recall/Sensitivity	0.947	**0.950**	0.974	**0.966**
F1-score	0.960	**0.958**	0.961	**0.958**
Number of Transcripts	2500		2499	

For human, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.11 :

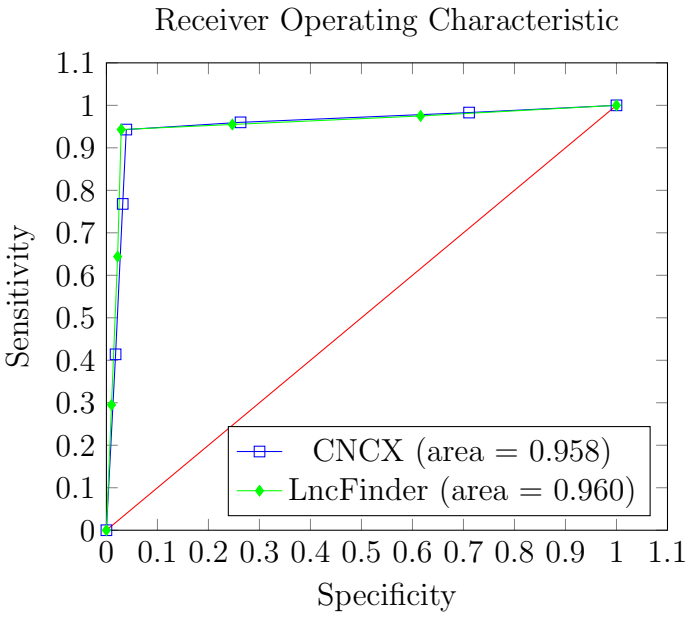

FIGURE 5.11: LncFinder and CNCX Comparison ROC Curve on human Test Data Set on Training Data Set 3

5.7.2 Test Data Set : Mouse (*Mus musculus*)

The *Accuracy* of LncFinder and CNCX are 92.53% and **93.81%**.

For mouse, the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.14 :

TABLE 5.14: Comparison between CNCX and LncFinder on Test Data Mouse

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.927	**0.923**	0.923	**0.954**
Recall/Sensitivity	0.923	**0.956**	0.928	**0.921**
F1-score	0.925	**0.939**	0.925	**0.937**
Number of Transcripts	1800		1800	

For mouse, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.12 :

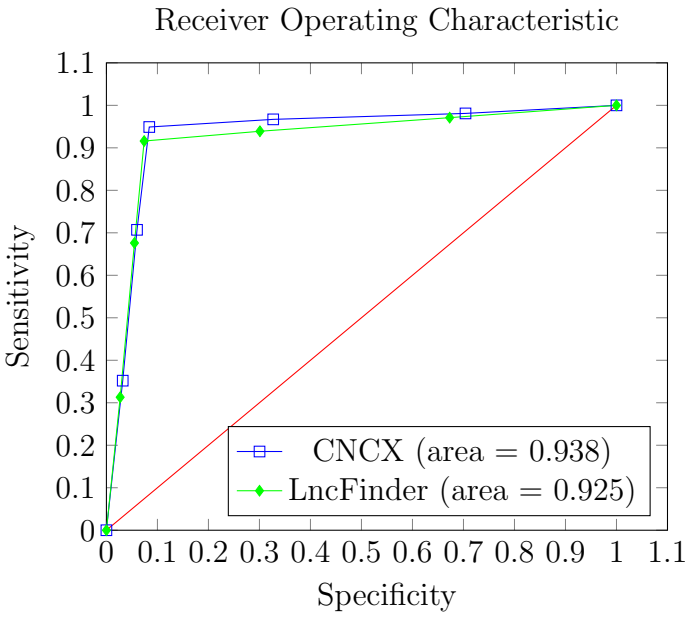

FIGURE 5.12: LncFinder and CNCX Comparison ROC Curve on Mouse Test Data Set on Training Set 3

5.7.3 Test Data Set : Zebrafish (*Danio rerio*)

The *Accuracy* of LncFinder and CNCX are 91.40% and **86.22%**.

For zebrafish, the comparison between **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.15 :

TABLE 5.15: Comparison between CNCX and LncFinder on Test Data Zebrafish

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.926	**0.894**	0.903	**0.835**
Recall/Sensitivity	0.900	**0.821**	0.928	**0.903**
F1-score	0.913	**0.856**	0.915	**0.868**
Number of Transcripts	3991		3991	

For zebrafish, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.13 :

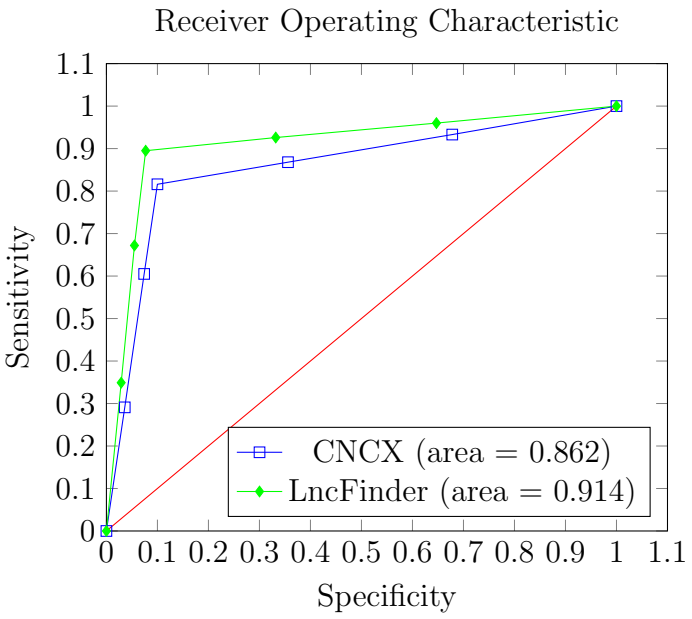

FIGURE 5.13: LncFinder and CNCX Comparison ROC Curve on Zebrafish Test Data Set on Training Data Set 3

5.7.4 Test Data Set : GENCODE version 28 Human

The *Accuracy* of LncFinder and CNCX are 91.02% and **92.51%**.

For GENCODE v28 human, the comparison between **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.16 :

TABLE 5.16: Comparison of CNCX and LncFinder on GENCODE v28 Human Data Set

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.880	**0.908**	0.945	**0.943**
Recall/Sensitivity	0.950	**0.946**	0.71	**0.905**
F1-score	0.914	**0.927**	0.907	**0.924**
Number of Transcripts	28467		28467	

For GENCODE v28 human, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.14 :

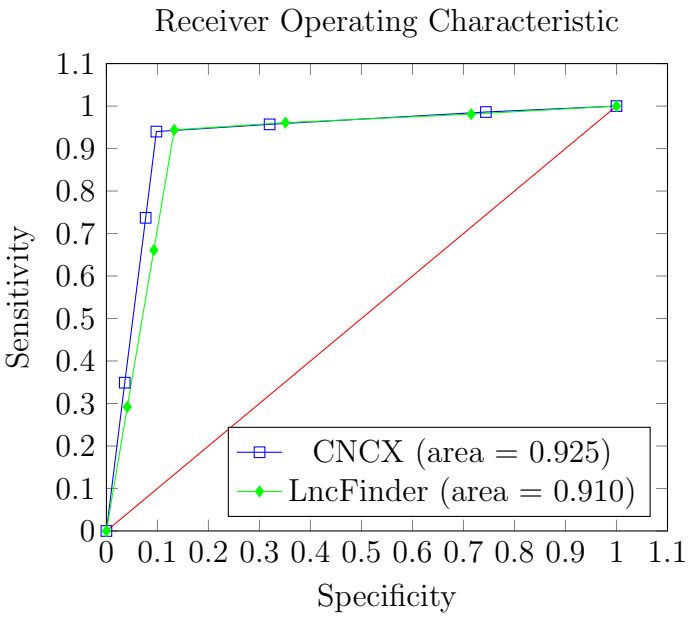

FIGURE 5.14: LncFinder and CNCX Comparison ROC Curve on GENCODE v28 Human Species on Training Data Set 3

5.7.5 Test Data Set : GENCODE version 18 Mouse

The *Accuracy* of LncFinder and CNCX are 92.22% and **96.09%**.

For GENCODE v18 mouse, the comparison among **CNCX** and **LncFinder** based on the **classification report** can be observed from table 5.17 :

TABLE 5.17: Comparison of CNCX and LncFinder on GENCODE v18 Mouse

Criteria	**Coding**		**Noncoding**	
	LncFinader	**CNCX**	**LncFinader**	**CNCX**
Precision	0.908	**0.948**	0.937	**0.974**
Recall/Sensitivity	0.905	**0.975**	0.939	**0.947**
F1-score	0.921	**0.961**	0.923	**0.960**
Number of Transcripts	36130		36130	

For GENCODE v18 mouse, the comparison between **CNCX** and **LncFinder** based on the **ROC characteristics** can be observed from graph 5.15 :

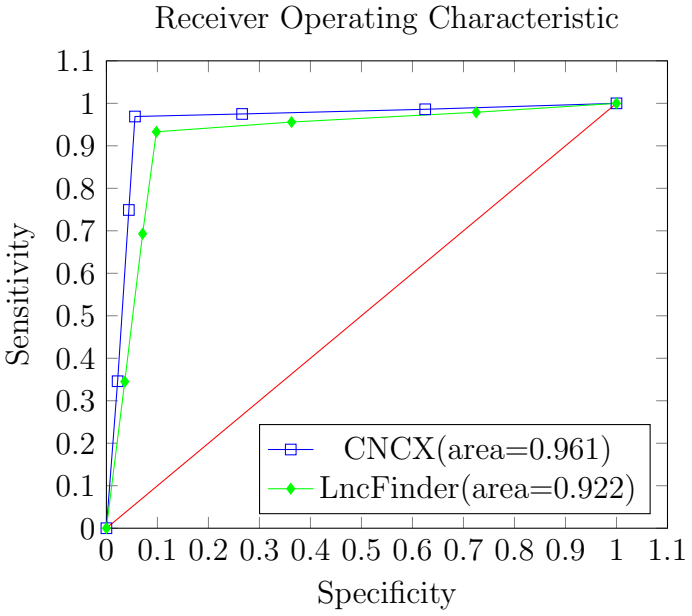

FIGURE 5.15: LncFinder and CNCX Comparison ROC Curve on GENCODE v18 Mouse Species on Training Data Set 3

5.8 Conclusion

The above discussion implies that *CNCX* can be very efficient in independent data sets where a data set contains RNA transcripts that need to be classified into coding or noncoding. *CNCX* outperforms *CPC* and *CPC2* in almost every field in all the test cases and *LncFinder* in some of the test cases.

The use of XGBoost algorithm has ensured control on over fitting, minimum loss and over specialization. This is how XGBoost has enabled superior performance of *CNCX*.

Chapter 6

Conclusions

6.1 Summary of Research

In this project, a new approach for the classification of coding and noncoding transcripts have been proposed. The focus was given in choosing features that are optimal and more biologically significant. The classifier model uses an advanced machine learning algorithm called eXtreme Gradient Boosting[5](XGBoost), which to the best of our knowledge has been first used in this project for biological sequence analysis. Experimental evaluation has been conducted on a group of data sets containing coding and noncoding transcripts of human and mouse genome and also collected data sets from CPC2[15], Lncfinder[11] and GENCODE[8].

The proposed method uses 14 features selected from renowned peer literature and selects 5 features among them by applying recursive feature elimination and chi-square feature selection method. The proposed *CNCX* method has performed better than *CPC* and *CPC2* in every testing scenario. In comparison to state of the art method *LncFinder*, CNCX has shown good promise by outperforming *LncFinder* in half of the testing scenarios and producing quite similar results for the rest of the scenarios.

6.2 Future Work

CNCX can classify protein coding and noncoding transcripts with high accuracy and so encourages further research as it is essential to solve this classification problem with 100% accuracy to proceed the knowledge of human biology. Some possible future work is summerized below:

- CNCX has been trained with 3 different data sets and tested on 5 different data sets of 3 different species. To train and test the model extensively with more real life data sets is one of the future plans. Only the boosting parameters have been tuned in case of parameter optimization. Optimizing other parameters to improve the model is another plan to be implemented in the future.

- Though manually adding the DNA-based feature(Fickett score) improves the model's accuracy, improving the feature selection so that Fickett score is present in the refined feature set methodically is also another future plan.

- Furthermore, the plan is to make this model more efficient altogether. It is hoped that those who want to start working on sequence classification based on XGBoost, this project will lead them to a good direction.

Bibliography

[1] *Parameter tuning tutorial*.

[2] Rujira Achawanantakun, Jiao Chen, Yanni Sun, and Yuan Zhang. Lncrna-id: Long non-coding rna identification using balanced random forests. *Bioinformatics*, 31(24):3897–3905, 2015.

[3] Junghwan Baek, Byunghan Lee, Sunyoung Kwon, and Sungroh Yoon. lncrnanet: Long non-coding rna identification using deep learning. *Bioinformatics*, 1:9, 2018.

[4] Rainer Breitling, Patrick Armengaud, Anna Amtmann, and Pawel Herzyk. Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS letters*, 573(1-3):83–92, 2004.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[6] Sean R Eddy. Non–coding rna genes and the modern rna world. *Nature Reviews Genetics*, 2(12):919, 2001.

[7] Manel Esteller. Non-coding rnas in human disease. *Nature Reviews Genetics*, 12(12):861, 2011.

[8] Adam Frankish, Mark Diekhans, Anne-Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M Mudge, Cristina Sisu, C Wright, Joel

Armstrong, et al. Gencode reference annotation for the human and mouse genomes. *Nucleic Acids Research*, 2018.

[9] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] Mitchell Guttman, Ido Amit, Manuel Garber, Courtney French, Michael F Lin, David Feldser, Maite Huarte, Or Zuk, Bryce W Carey, John P Cassady, et al. Chromatin signature reveals over a thousand highly conserved large non-coding rnas in mammals. *Nature*, 458(7235):223, 2009.

[11] Siyu Han, Yanchun Liang, Qin Ma, Yangyi Xu, Yu Zhang, Wei Du, Cankun Wang, and Ying Li. Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Briefings in bioinformatics*, 2018.

[12] Yong Huang, Quan Zou, Shun Ming Tang, Li Gang Wang, and Xing Jia Shen. Computational identification and characteristics of novel microrna s from the silkworm (bombyx mori l.). *Molecular biology reports*, 37(7):3171–3176, 2010.

[13] Nicholas T Ingolia, Gloria A Brar, Noam Stern-Ginossar, Michael S Harris, Gaëlle JS Talhouarne, Sarah E Jackson, Mark R Wills, and Jonathan S Weissman. Ribosome profiling reveals pervasive translation outside of annotated protein-coding genes. *Cell reports*, 8(5):1365–1379, 2014.

[14] Matthew K Iyer, Yashar S Niknafs, Rohit Malik, Udit Singhal, Anirban Sahu, Yasuyuki Hosono, Terrence R Barrette, John R Prensner, Joseph R Evans, Shuang Zhao, et al. The landscape of long noncoding rnas in the human transcriptome. *Nature genetics*, 47(3):199, 2015.

[15] Yu-Jian Kang, De-Chang Yang, Lei Kong, Mei Hou, Yu-Qi Meng, Liping Wei, and Ge Gao. Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic acids research*, 45(W1):W12–W16, 2017.

[16] Philipp Kapranov and Georges St Laurent. Dark matter rna: existence, function, and controversy. *Frontiers in genetics*, 3:60, 2012.

[17] Lei Kong, Yong Zhang, Zhi-Qiang Ye, Xiao-Qiao Liu, Shu-Qi Zhao, Liping Wei, and Ge Gao. Cpc: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic acids research*, 35(suppl_2):W345–W349, 2007.

[18] Aimin Li, Junying Zhang, and Zhongyin Zhou. Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme. *BMC bioinformatics*, 15(1):311, 2014.

[19] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[20] Cong Pian, Guangle Zhang, Zhi Chen, Yuanyuan Chen, Jin Zhang, Tao Yang, and Liangyun Zhang. Lncrnapred: Classification of long non-coding rnas and protein-coding transcripts by the ensemble algorithm with a new hybrid feature. *PloS one*, 11(5):e0154567, 2016.

[21] Chris P Ponting, Peter L Oliver, and Wolf Reik. Evolution and functions of long noncoding rnas. *Cell*, 136(4):629–641, 2009.

[22] John R Prensner and Arul M Chinnaiyan. The emergence of lncrnas in cancer biology. *Cancer discovery*, 1(5):391–407, 2011.

[23] Man-Tang Qiu, Jing-Wen Hu, Rong Yin, and Lin Xu. Long noncoding rna: an emerging paradigm of cancer research. *Tumor Biology*, 34(2):613–620, 2013.

[24] Kun Sun, Xiaona Chen, Peiyong Jiang, Xiaofeng Song, Huating Wang, and Hao Sun. iseerna: identification of long intergenic non-coding rna transcripts from transcriptome sequencing data. *BMC genomics*, 14(2):S7, 2013.

[25] Lei Sun, Hui Liu, Lin Zhang, and Jia Meng. lncrscan-svm: a tool for predicting long non-coding rnas using support vector machine. *PloS one*, 10(10):e0139654, 2015.

[26] Liang Sun, Haitao Luo, Dechao Bu, Guoguang Zhao, Kuntao Yu, Changhai Zhang, Yuanning Liu, Runsheng Chen, and Yi Zhao. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic acids research*, 41(17):e166–e166, 2013.

[27] Ryan J Taft, Ken C Pang, Timothy R Mercer, Marcel Dinger, and John S Mattick. Non-coding rnas: regulators of disease. *The Journal of pathology*, 220(2):126–139, 2010.

[28] Rashmi Tripathi, Sunil Patel, Vandana Kumari, Pavan Chakraborty, and Pritish Kumar Varadwaj. Deeplnc, a long non-coding rna prediction tool using deep neural network. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):21, 2016.

[29] Liguo Wang, Hyun Jung Park, Surendra Dasari, Shengqin Wang, Jean-Pierre Kocher, and Wei Li. Cpat: Coding-potential assessment tool using an alignment-free logistic regression model. *Nucleic acids research*, 41(6):e74–e74, 2013.

[30] Xiaolin Zhou and Jie Xu. Identification of alzheimer's disease–associated long noncoding rnas. *Neurobiology of aging*, 36(11):2925–2931, 2015.

List of Notations

CNCX - Classification of Coding and Noncoding Transcripts Based on eXtreme Gradient Boosting Classifier

XGBoost - eXtreme Gradient Boosting

DNA - Deoxyribonucleic acid

RNA - Ribonucleic Acid

Transcript - A transcript is the single-stranded RNA product synthesized by transcription of DNA

ncRNA - Noncoding RNA

lncRNA - Long noncoding RNA

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

TPR - True Positive Rate/Sensitivity

FPR - False Positive Rate

PPV - Positive Predictive Value/Precision