Universiteit Leiden

Master Computer Science

LncRNA-BERT: An RNA Language Model for Classifying Coding and Long Non-Coding RNA

Name:	Luuk Romeijn
Student ID:	2592800
Date:	27/01/2025
Specialisation:	Bioinformatics
1st supervisor:	Katy Wolstencroft
2nd supervisor:	Hailiang Mei

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

RNA transcripts generated in Next Generation Sequencing experiments require accurate coding potential classification, given the increasing evidence that long non-coding RNAs (lncRNAs) play crucial regulatory roles. Despite the availability of a wide variety of lncRNA classifiers, none of the purely sequence-based algorithms proposed in previous works outperform the most advanced feature-based approaches. We present lncRNA-BERT, an RNA language model pre-trained on 0.5 million human mRNA/lncRNA sequences that achieves state-of-the-art performance in classifying coding and long non-coding RNA. A large collection of features proposed in previous works is used to demonstrate the potential of an RNA language model for this task, while also establishing a random forest baseline model with outstanding performance. The pre-trained lncRNA-BERT model is shown to generate biologically relevant embeddings that distinguish mRNA from lncRNA without supervised learning, confirming that coding potential is a sequence-intrinsic characteristic. We compare lncRNA-BERT to other Nucleotide Language Models and demonstrate the benefit of pre-training on human data compared to the commonly used RNAcentral dataset. In addition, our novel Convolutional Sequence Encoding method is shown to be more effective and efficient than K-mer Tokenization and Byte Pair Encoding for pre-training on long sequences that are otherwise above the common context size.

The methods and results presented in this thesis were generated as part of an internship project at the Sequencing Analysis Support Core (SASC) at Leiden University Medical Center (LUMC).

A paper version of this thesis (available through pre-print: Romeijn, Cats, et al. 2025) has been submitted to the journal: 'RNA Biology'.

Acknowledgements

There are several people that I owe a special word of thanks to. Firstly, I would like to thank Leon Mei from the Sequencing Analysis Support Core (SASC) at LUMC for offering me this internship opportunity. I really appreciated your responsiveness, collaboration, and extensive guidance. Secondly, I would like to thank Katy Wolstencroft from LIACS, for providing feedback from a birds-eye view, which is especially valuable for someone who tends to get caught up in specific details. Also, a special thanks to the other interns and employees at SASC, for letting me join the daily scrum and for being good company during office hours. Finally, I would like to express my gratitude towards my family, friends, and anyone who made my daily life during my master's program a bit easier. Thank you for showing me that, even though I study computers, it is being human that matters most.

Contents

1 Introduction **1**
- 1.1 The Definition and Relevance of Long Non-Coding RNA . . . 1
- 1.2 Limitations of LncRNA Classifiers for Novel Transcripts Detected in NGS . . . 1
- 1.3 Developments in Nucleotide Language Models . . . 3
- 1.4 Scope of This Thesis . . . 3

2 Related Work **5**
- 2.1 LncRNA Classifiers . . . 5
- 2.2 Nucleotide Language Models . . . 11

3 Methods **15**
- 3.1 Data . . . 15
- 3.2 Feature-Based Approach . . . 17
- 3.3 Encoding Methods . . . 18
- 3.4 Neural Architecture . . . 22
- 3.5 Training . . . 23
- 3.6 Experimental Setup . . . 24

4 Results **25**
- 4.1 Comparison to Established LncRNA Classifiers . . . 25
- 4.2 Feature-Based Exploratory Data Analysis and Classification Baselines . . . 25
- 4.3 Pre-Training . . . 29
- 4.4 Encoding Methods . . . 31
- 4.5 Comparison to Existing NLMs . . . 35
- 4.6 Latent Space Inspection . . . 37

5 Discussion **41**
- 5.1 Benefits and Limitations of NLMs for LncRNA Classification . . . 42
- 5.2 Data Clustering Holds Potential For Future Work . . . 42
- 5.3 Addressing the Competitiveness of LncRNA-BERT with Other NLMs . . . 43
- 5.4 Recommendations for Improving NLMs . . . 43

A Supplementary Information on Feature-Based Approach **51**
- A.1 List of Features Included in Analysis . . . 51
- A.2 LncRNA-LR and LncRNA-RF Feature Importance . . . 51
- A.3 Validating the CPAT Re-Implementation . . . 51

B LncRNA Classifier Comparison Table **55**

C Hyperparameter Tuning **57**

Chapter 1

Introduction

1.1 The Definition and Relevance of Long Non-Coding RNA

The central dogma of molecular biology (Crick 1958) is often misinterpreted as the inevitable conversion of a gene into a functioning protein. Contradicting this misconception is the fact that the majority (76–97%) of the human genome encodes for RNA molecules that do not translate into protein products (Nemeth et al. 2023). While initially overlooked, these non-coding RNAs (ncRNAs) are now known to have important regulatory functions. For most types of short RNAs, we also know their mode of action. For example, microRNAs (miRNAs) inhibit gene expression by binding to the 3' untranslated region (UTR) of coding messenger RNA (mRNA), and Piwi-interacting RNA (piRNA) can induce the silencing of transposons by binding to Piwi proteins (Ender et al. 2010).

On the other hand, the functions and mechanisms of most long ($\geq$200 nt) non-coding RNAs (lncRNAs) remain largely unknown (Mattick et al. 2023). In fact, the lack of a better categorization for these transcripts explains why they are simply labelled as 'lncRNA' in most literature. Despite that, lncRNAs are a highly prevalent type of RNA, underlined by the 173,112 human lncRNA transcipts stored in the NONCODE (v6) database versus the 197,151 mRNAs in RefSeq (r225).

While a systematic characterization of lncRNAs does not yet exist, an increasing number of studies show that lncRNAs have significant and highly diverse regulatory functions across various genomic levels. As depicted in Figure 1.1, specific lncRNAs can interact with DNA, RNA, and/or proteins (Nemeth et al. 2023). Such interactions allow them to control chromatin architecture, modulate enhancer activity, and modify the composition of biomolecular condensates (large protein-RNA complexes) in the cell (Mattick et al. 2023).

Because of their regulatory functions, lncRNAs are often associated with disease. Notable examples of such lncRNAs are LINCMD1 and HOTAIR. LINCMD1 acts as a miRNA competitor for muscle-specific transcription factors, hence its expression levels can be related to Duchenne Muscular Dystrophy (Cesana et al. 2011). HOTAIR regulates cell proliferation by interacting with the YBX1 protein, resulting in an association with several cancer types (S. Li et al. 2021). These examples indicate that studying lncRNAs can lead to an improved understanding of disease phenotypes, which may help in the development of drug therapies.

1.2 Limitations of LncRNA Classifiers for Novel Transcripts Detected in NGS

Predicting whether a novel transcript is coding or non-coding is an important step in many Next Generation Sequencing (NGS) pipelines. The output of this task, from hereon referred to as 'lncRNA classification', provides researchers with crucial context to newly discovered, unannotated RNA sequences, often identified through RNA-Seq experiments (e.g. Barriocanal et al. 2015; C. Fan et al. 2023; Weikard et al. 2013).

Nowadays, researchers can choose between over 40 lncRNA classification algorithms. Most of these classifiers are Machine Learning (ML) models that have been trained on annotated RNA data from resources such as RefSeq and GENCODE. Famous examples of these methods are CPC, CNCI, CPAT, and CPC2 (Y.-J. Kang et al. 2017; Kong et al. 2007; Liang Sun et al. 2013; L. Wang et al. 2013). These algorithms are more advanced than simply detecting the presence of an Open Reading Frame (ORF),

as many lncRNAs contain short ORFs, which are sometimes translated into important protein products (Figure 1.1e, Pang et al. 2018). Non-coding isoforms of mRNAs further complicate the distinction between the two RNA types (Nam et al. 2016). The difficulty and inherent ambiguity of this classification problem stimulates continuous development of novel lncRNA classifiers, leveraging the latest available annotations and methodological advancements.

The majority of lncRNA classification methods use a set predefined sequence features such as ORF length, k-mer frequencies, and isolectric point that can serve as predictors for coding potential. Early lncRNA classifiers were mostly based on support vector machines (Kong et al. 2007; Liang Sun et al. 2013), whereas recent algorithms are powered by boosting models or neural networks (Camargo et al. 2020; Feng et al. 2023). Data resources such as RefSeq and GENCODE are commonly used as training input. Additionally, some studies increase performance by maximizing data diversity (Feng et al. 2023; S. Liu et al. 2019) or by carrying out feature selection procedures (Han, Liang, Ma, et al. 2018; Y.-J. Kang et al. 2017; Yu Zhang et al. 2020).

Multiple benchmarking studies have shown that the performance of these existing methods is quite good: top models can achieve an F1-score of 93-96% on independent test sets (Han, Liang, Y. Li, et al. 2016b; D. Singh et al. 2022; Zheng et al. 2021). However, while a model that relies on a predetermined set of predictors may learn the relationship between these features and the target, it may fail to capture the true underlying signal. Purely sequence-based deep learning methods for lncRNA classification have been proposed in previous works (Baek et al. 2018; Hill et al. 2018; Meng, Q. Kang, et al. 2021), but these were outperformed by feature-based or hybrid methods in a recent benchmark (D. Singh et al. 2022). These approaches utilized convolutional and/or recurrent neural networks, which suffer from limited receptive fields and exploding/vanishing gradients, respectively. The transformer architecture improves upon these designs by employing an attention mechanism and being non-recurrent (Vaswani et al. 2017). We demonstrate that this more advanced architecture can improve sequence-based lncRNA classifiers.

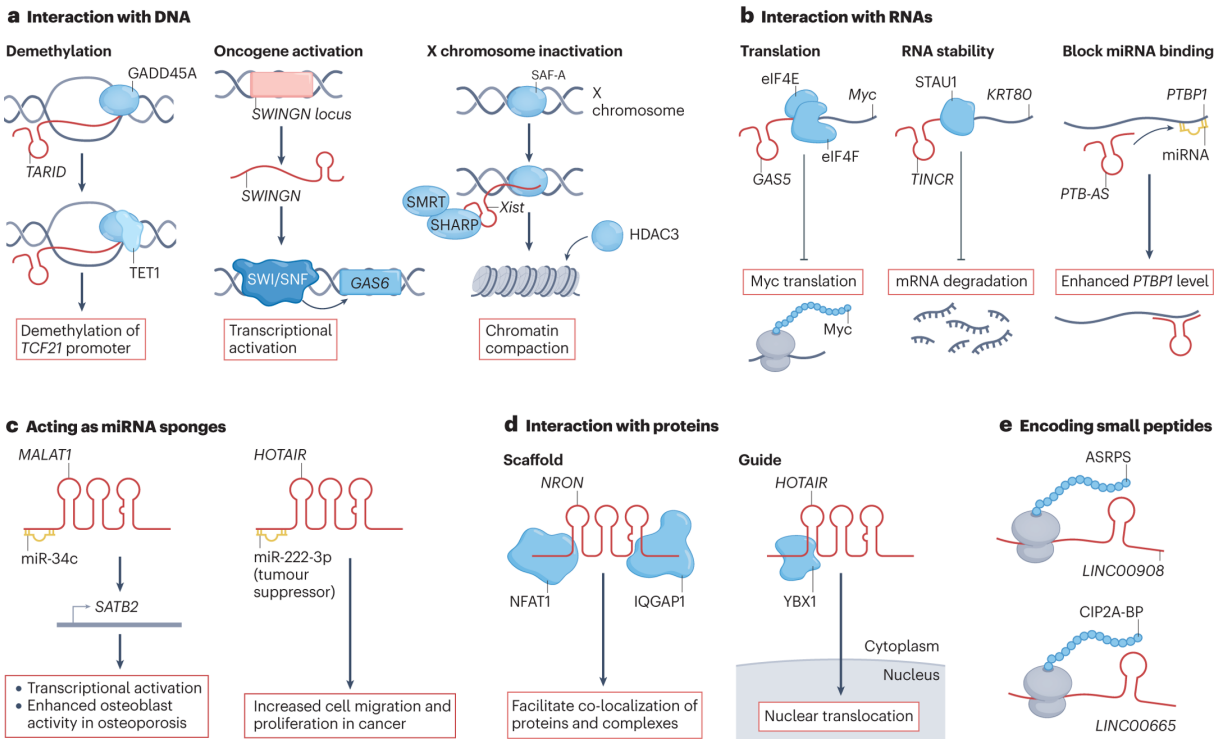

Figure 1.1: The main functions of lncRNAs (Nemeth et al. 2023, Figure 2) a) Several lncRNAs were found to interact with DNA, modulating enhancer activity or causing epigenetic modifications that influence gene expression. b) Some lncRNAs affect the translation levels of mRNAs through interactions with mRNA or RNA-bound protein complexes. c) LncRNAs (e.g. HOTAIR) can act as miRNA sponges, causing the upregulation of certain genes. d) Specific lncRNAs bind proteins and act as scaffolds or guides. e) lncRNAs may contain short ORFs that encode for small, functional peptides.

1.3 Developments in Nucleotide Language Models

Nucleotide Language Models (NLMs) are emerging as novel analysis tools for DNA/RNA data, capable of various downstream tasks such as variant prioritization, splice site detection, and secondary structure prediction of RNA (Dalla-Torre et al. 2023; Penić et al. 2024). Their development is stimulated by the success of the transformer-based Large Language Models (LLMs) in Natural Language Processing (NLP) and various other domains. NLMs are usually presented as general-purpose foundation models: they are pre-trained on large collections of nucleotide data with the Masked Language Modeling (MLM) objective to generate informative sequence embeddings.

Different types of NLMs are available, specializing in DNA or RNA based on their pre-training data. Examples of DNA language models are DNABERT-2 (Z. Zhou, Ji, et al. 2023), GENA-LM (Fishman et al. 2023), and Nucleotide Transformer (Dalla-Torre et al. 2023). RNA language models include RNA-FM (Chen et al. 2022), RNAErnie (N. Wang et al. 2024), and RiNALMo (Penić et al. 2024). DNA LMs are usually pre-trained on genomic data. For RNA LMs, the most popular resource is RNAcentral, containing mostly short non-coding RNA of multiple species. We anticipate that using data from resources like GENCODE and RefSeq can improve the performance of RNA LMs on mRNA and lncRNA.

Another crucial component of an NLM is its encoding method, which determines the type of linguistic units (tokens) that the model receives as input. Different encoding methods affect the model in its maximally accepted input length, resolution, and learned capabilities. So far, Nucleotide-Level Tokenization (Akiyama et al. 2022), K-mer Tokenization (Dalla-Torre et al. 2023), and Byte Pair Encoding (Z. Zhou, Ji, et al. 2023) have been proposed, each having its own set of (dis)advantages in one or more of the aforementioned aspects. Increasing model/data size and applying the latest architectural advancements has led NLMs to where they are now in terms of performance (Dalla-Torre et al. 2023; Fishman et al. 2023; Penić et al. 2024). Nevertheless, we wonder whether the encoding methods used by these models truly reflect the nature of DNA and present a novel technique in this study.

1.4 Scope of This Thesis

In this thesis, we propose lncRNA-BERT (Long Non-Coding RNA Bidirectional Encoder Representations from Transfomers), an RNA language model pre-trained on human mRNA/lncRNA data and fine-tuned for sequence-based lncRNA classification. LncRNA-BERT utilizes a novel Convolutional Sequence Encoding (CSE) technique. Specifically, we address the following Research Questions:

1. How accurately can lncRNA-BERT distinguish coding from long non-coding RNA in comparison to existing methods?
2. To what extent do previous lncRNA classifiers and their respective coding potential predictor features motivate the use of an RNA language model for lncRNA classification?
3. Does an RNA language model for lncRNA classification benefit from pre-training, specifically on human data?
4. Can we design a more effective and efficient sequence encoding method for long nucleotide sequences?

By addressing these research questions, we aim to develop an improved sequence-based lncRNA classification method, while also assessing its biological relevance and limitations. In doing so, we aim to gain an improved understanding of lncRNA and further advance the development of NLMs.

Chapter 2

Related Work

An extensive collection of scientific research precedes our approach to lncRNA classification. On the one hand, a significant body of research (40+ studies) is dedicated to the advancement of lncRNA classifiers. Section 2.1 serves as an overview of these algorithms. On the other hand, specific to our approach, recent years have witnessed a growing amount of research on Nucleotide Language Models (NLMs), described in Section 2.2.

2.1 LncRNA Classifiers

Numerous prior studies have investigated how pcRNAs can be distinguished from lncRNAs. These studies differ in 1) what kind of features are used as predictory variables; 2) the datasets that are used for training; and 3) the machine/deep learning models and methods that are applied. We shall addresses these three components in this order. An extensive overview is given in Table 2.2.

2.1.1 Features

A wide range of sequence-derived features have been proposed as possible coding-potential predictors in previous works. Similar to (Han, Liang, Ma, et al. 2018; J. Li et al. 2020; Zheng et al. 2021), we provide a categorization of these features. Six main feature types are identified and listed in Table 2.1. Note that a sequence's Open Reading Frame (ORF) and other patterns are purely sequence-intrinsic, and most physicochemical and secondary structure features can be indirectly inferred from sequence data. Hence, these features motivate the use of an NLM w.r.t. Research Question 2. Nevertheless, numerous methods enrich these sequence-intrinsic features with extrinsic data, for example through database alignments and genome mapping. This provides these feature-based algorithms with data that cannot be learned by NLMs.

	Feature type	**Example**	**Relevance**
1	ORF	ORF length	PcRNAs are more likely to consist of longer ORFs than lncRNAs.
2	Sequence patterns	K-mer frequencies, Fickett score	Some (combinations) of nucleotides appear more often in pcRNA than in lncRNA.
3	Database alignment	BLASTX hits	PcRNAs will yield more protein database hits than lncRNAs.
4	Genome mapping	Conservation score	Some features can only or more efficiently be deduced from a reference genome.
5	Physicochemical	Isoelectric point	PcRNAs can exhibit different physicochemical properties than lncRNAs.
6	Secondary structure	Unpaired-paired bases frequency	The secondary structure can aid in determining how likely an RNA is to be protein-coding.

Table 2.1: The six main feature types as used by existing lncRNA classification methods.

Algorithm	Reference	Type	Model	Feature type						Data source	Citations
				1	2	3	4	5	6		
CPC	Kong et al. 2007	ML	SVM	x		x				NONCODE, RNAdb, SwissProt	2028
CNCI	Liang Sun et al. 2013	ML	SVM		x					GENCODE, NONCODE	1304
CPAT	L. Wang et al. 2013	ML	Logistic regression	x	x					RefSeq, prior study	1275
CPC2	Y.-J. Kang et al. 2017	ML	SVM	x	x			x		RefSeq, prior study	768
PhyloCSF	Lin et al. 2011	SM	Phylogeny log ratio							Prior study	715
PLEK	A. Li et al. 2014	ML	SVM		x					GENCODE, RefSeq	483
FEELnc	Wucher et al. 2017	ML	Random forest	x	x					GENCODE	269
iSeeRNA	K. Sun et al. 2013	ML	SVM	x	x		x			HAVANA	130
CONC	J. Liu et al. 2006	ML	SVM		x	x		x	x	NONCODE, RNAdb, SwissProt	113
LncFinder	Han, Liang, Ma, et al. 2018	ML	SVM	x	x			x	x	GENCODE	88
CNIT	Guo et al. 2019	ML	XGBoost		x					RefSeq	87
LncADeep	C. Yang et al. 2018	DL	DBN	x	x	x				GENCODE, RefSeq	86
COME	Hu et al. 2016	ML	Random forest		x		x		x	GENCODE	81
lncRScan-SVM	Lei Sun et al. 2015	ML	SVM		x		x			GENCODE	81
lncRNA-MFDL	X.-N. Fan and S.-W. Zhang 2015	DL	DSN	x	x				x	GENCODE, RefSeq	78
PLncPro	U. Singh et al. 2017	ML	Random forest	x	x	x				CANTATAdb	76
LncRNA-ID	Achawanantakun et al. 2015	ML	Random forest	x	x	x				GENCODE	76
CPPred	Tong et al. 2019	ML	SVM	x	x			x		Ensembl, RefSeq	69
PORTRAIT	Arrial et al. 2009	ML	SVM	x	x			x		NONCODE, Rfam, RNAdb, SwissProt	69
LncRNAnet	Baek et al. 2018	DL	CNN/RNN							GENCODE	63
LGC	G. Wang et al. 2019	SM		x	x					GENCODE, RefSeq	54
mRNN	Hill et al. 2018	DL	RNN							GENCODE	50
LncRNApred	Pian et al. 2016	ML	Random Forest	x	x					NONCODE, UCSC	43
RNAplonc	Negri et al. 2018	ML	Decision Tree		x				x	GreeNC, Phytozome, PLNlncRbase	42
CREMA	Simopoulos et al. 2018	ML	Random Forest	x	x	x				Ensembl, RefSeq	41
longdist	Schneider et al. 2017	ML	SVM	x	x					Ensembl, GENCODE	40
RNAsamba	Camargo et al. 2020	DL	IGLOO	x	x					GENCODE, RefSeq	39
DeepCPP	Yu Zhang et al. 2020	DL	CNN	x	x					Ensembl, RefSeq	35
lncScore	J. Zhao et al. 2016	ML	Logistic Regression	x	x					GENCODE	32
BASiNET	Ito et al. 2018	ML	Decision Tree		x					GENCODE, RefSeq	31
PredLNC-GFStack	S. Liu et al. 2019	ML	Random Forest	x	x			x		GENCODE	18
PlncRNA-HDeep	Meng, Q. Kang, et al. 2021	DL	RNN/CNN							GreeNC, RefSeq	14
lncRNA_Mdeep	X.-N. Fan, S.-W. Zhang, et al. 2020	DL	DNN/CNN	x	x			x		CANTATAdb, Ensembl	13
NCResNet	S. Yang et al. 2020	DL	DNN	x	x			x	x	Ensembl, RefSeq	10
PreLNC	Cao et al. 2020	ML	Random Forest	x	x			x		Ensembl, GreeNC	7
Lncident	Han, Liang, Y. Li, et al. 2016a	ML	SVM	x	x					GENCODE	5
lncRNA-LSTM	Meng, Chang, et al. 2019	DL	LSTM							GreeNC	4
LncDLSM	Y. Wang et al. 2023	DL	CNN		x					NONCODE, RefSeq	2
LncDC	M. Li et al. 2022	ML	XGBoost	x	x			x	x	GENCODE, RefSeq	2
LncCat	Feng et al. 2023	ML/DL	CatBoost, BERT	x	x					Ensembl, GENCODE, RefSeq	2

Table 2.2: Overview of lncRNA classification methods presented in prior works. We categorizes the applied methodology in one of three types: Machine Learning (ML), Deep Learning (DL), or Statistical Model (SM), where statistical model defines a statistical model that is tailor-made for the task at hand. Feature types: 1=ORF; 2=Sequence patterns; 3=Database alignment; 4=Genome mapping; 5=Physicochemical; 6=Secondary structure. Data source: main training sources are reported. If training sets for multiple species are used, reported human training sets. Citations: according to Web Of Science, retrieved on 21-5-2024.

2.1.1.1 Open Reading Frame

The most simple example of a feature used by lncRNA classifiers is the length of a transcript's longest ORF. While introduced in 2007 by an early coding potential tool called CPC (Kong et al. 2007), the ORF length is a popular explanatory variable, even in more recent studies (Camargo et al. 2020; Yu Zhang et al. 2020). The intuition behind this feature is simple: transcripts with long ORFs are likely to be protein-coding. The same applies for ORF coverage, where pcRNAs are expected to be mostly covered by an ORF. Note that the presence of ORFs in a transcript alone is not sufficient for distinguishing between pcRNAs and lncRNAs, as lncRNAs are known to have short ORFs (Section 4.2.1), which may encode for functional micro-peptides (Pang et al. 2018).

The applied ORF identification method and the quality and completeness of a transcript are of crucial influence in whether or not an ORF is found within a sequence. Hence, FEELnc introduces 5 alternative definitions, each with different selection criteria (Wucher et al. 2017). The most strict variant, in which both a start and stop codon must be present, is most often used in the literature and also in NCBI's online ORFfinder tool (`https://www.ncbi.nlm.nih.gov/orffinder`). Other definitions proposed in that study are more relaxed, allowing for missing start and/or stop codons, or even reverting back to the full transcript in case of a lack of both. A downside to this method is that higher relaxation levels increase the chance of false positives. Alternatively, a different work successfully applied a CNN for the identification of ORFs (Baek et al. 2018), allowing for a data-driven identification of ORFs beyond hard-coded rules.

2.1.1.2 Sequence Patterns

A second and widely used class of discriminative features comprises nucleotide patterns, or specifically the occurrence bias of certain patterns in pcRNA compared to lncRNA. This bias can be expressed in terms of pattern frequencies, scoring systems, distance measures, sequence distribution, and identifying most-like coding sequences.

2.1.1.2.1 Pattern frequencies

The most straightforward way of representing sequence bias is by counting pattern frequency. CONC was the first method to incorporate monomer, dimer, and trimer nucleotide frequencies (J. Liu et al. 2006). FEELnc developed a fast k-mer counting algorithm to enable the inclusion of 12-mers (Wucher et al. 2017). PLEK introduced a normalization scheme, multiplying k-mer frequencies by a factor $w = 1/(4^{5-k})$ to correct for high probabilities of short k-mers (A. Li et al. 2014). Another example is LncFinder, which applies k-mer frequencies to the ORF (Han, Liang, Ma, et al. 2018). Assuming a correct reading frame allows LncFinder to count codons, using a step size of $k = 3$. Finally, DeepCPP calculates the frequencies of discontinuous k-mers, which was shown to lead to an increased performance (Yu Zhang et al. 2020).

2.1.1.2.2 Scoring systems

An alternative manner to utilize nucleotide pattern bias for sequence classification is through scoring systems, which assign higher scores to transcripts with subsequences that occur more often pcRNA than lncRNA. This way, k-mer frequencies can be summarized into a single value, reducing the number of covariates in a model. CPAT (L. Wang et al. 2013) was the first to propose the hexamer score, formulated as the mean log ratio $\frac{F_{pc}(W_i)}{F_{nc}(W_i)}$ of the occurrence frequencies of hexamers W_i in a sequence. The hexamer score and variations thereof were utilized by several later classifiers (Simopoulos et al. 2018; C. Yang et al. 2018; J. Zhao et al. 2016).

The Fickett testcode statistic (or Fickett score) adds another level to nucleotide pattern bias by adding a positional component (Fickett 1982; L. Wang et al. 2013), which reflects the extent to which a certain base is favored in a specific reading frame. Usage and position values are derived from look-up tables which have been calculated from a very small dataset of pcRNAs and ncRNAs in a study dating from 1982 (Fickett 1982). Hence, it is notable that many RNA classification methods successfully apply the Fickett score as a predictor for protein-coding capability (K. Sun et al. 2013; L. Wang et al. 2013; C. Yang et al. 2018).

DeepCPP (Yu Zhang et al. 2020) utilizes a new type of scoring system, assigning scores based only on nucleotides around the start codon of the identified ORF. The authors developed this metric based on biological findings that indicated the importance and conservation of bases in this area.

2.1.1.2.3 Distance measures

LncFinder calculates a distance measure between the k-mer frequency spectrum of a query and that of the average pcRNA or ncRNA in a training dataset (Han, Liang, Ma, et al. 2018), to be less reliant on pre-calculated log ratios e.g. used in the hexamer score. While the use of this feature was proven to be highly successful, it is based on a false intuition: lncRNAs are expected to be closer to the average lncRNA spectrum than pcRNAs. We show that this is not the case and demonstrate that LncFinder's distance measure is mostly an indication of sequence entropy (Section 4.2.3).

2.1.1.2.4 Sequence distribution

CPPred investigates the use of CTD (Composition, Transition, Distribution) features for lncRNA classification (Tong et al. 2019), from which the distribution feature is the most novel (composition and transition are similar to k-mer frequencies). This feature describes, for each base, the proportion of times that it occurs in the first 25%, 50%, 75%, and 100% of the sequence, relative to the sequence length. One of the outcomes of the work is that the proportion of T's in the first half of the sequence is a key feature for predicting the coding capability of a transcript. This hints at the presence of long-ranging nucleotide patterns, which features discussed so far fail to capture.

Another feature related to sequence distribution is based on the intuition that long ORFs cannot be interrupted by stop codons. In the case of a pcRNA, we should see a discrepancy between the number of stop codons in each reading frame, expecting the number of stop codons in one of these frames (that of the ORF) to be less than the others. Therefore, lncRScan-SVM proposes to use the standard deviation of the stop codon reading frame distribution as a feature for distinguishing coding from non-coding transcripts (Lei Sun et al. 2015).

2.1.1.2.5 Most-Like Coding Sequence

To address the challenges that come with identifying ORFs in transcripts, CNCI identifies a so-called Most-Like Coding Sequence (MLCDS) based purely on nucleotide pattern bias (Liang Sun et al. 2013). The procedure for identifying an MLCDS is described by the following steps: 1) calculating the usage frequency bias (log ratio) of Adjoined Nucleotide Triplets (ANTs); 2) generating six arrays of ANT scores based on the ANTs of six possible transcript reading frames; and finally 3) applying dynamic programming for identifying the subarray with the largest consecutive sum value. This yields six MLCDSs with different scores, corresponding to three reading frames in two possible directions. The functional relevance of the reverse direction is questionable in our opinion. RNA transcripts are single-stranded and sequenced from 5' to 3'. Furthermore, the authors do not mention reverse complementing the sequence, nor do they mention the calculation of the ANT usage bias for these reversed sequences. MLCDS is utilized by CNCI, CNIT, and LncADeep (Guo et al. 2019; Liang Sun et al. 2013; C. Yang et al. 2018).

2.1.1.3 Database Alignment

Some lncRNA classification tools search a protein reference database to identify whether RNAs encode for known protein products. CPC and PLncPro deploy BLASTX for a database search against Uniref90 and Swiss-Prot, respectively, using the number of hits as the main feature (Kong et al. 2007; U. Singh et al. 2017). Additionally, both studies reason that protein-coding transcripts should have higher quality matches that reside mostly in the same reading frame. Hereto, CPC and PLncPro aggregate the e-values of BLASTX hits into a single score, and also describe the variance or entropy of hits across different reading frames. Furthermore, PLncPro uses the total bit score as a final BLAST-derived feature. As an alternative to BLAST, LncADeep and LncRNA-ID deploy the Hidden Markov Model-based HMMER as a database search method (Achawanantakun et al. 2015; C. Yang et al. 2018), and CREMA uses DIAMOND as a faster BLAST alternative (Simopoulos et al. 2018).

Despite its proven discriminatory power, results of a database search are not often used as predictive features for lncRNA classifiers. Reasons for this are the computational complexity of algorithms like BLAST, and the dependence on a reference protein database. CPC's successor, CPC2, steps away from BLASTX to achieve a significant speed-up and to make the model more species-neutral (Y.-J. Kang et al. 2017). Nevertheless, BLASTX hits can serve as evidence for the classification of pcRNAs, which is not offered by CPC2 and other reference-free classifiers.

2.1.1.4 Genome Mapping

We define genome mapping as a fourth feature category, indicating features that have been derived from genomic rather than transcriptomic sequence data. In order to calculate such features, transcripts must be mapped or aligned to a reference genome. Consequently, algorithms with predictors in this category, like COME, iSeeRNA, and LncRScan-SVM (Hu et al. 2016; K. Sun et al. 2013; Lei Sun et al. 2015), require a GTF file with genome coordinates as input. The benefit of using genome-mapped transcript data is that it enables the inclusion of some additional features, as explained below.

The most common genome-derived feature is the DNA conservation score, derived from PhastCons annotations. PhastCons is a program that, given a multiple sequence alignment, determines how well certain parts of a sequence are conserved across evolution. Coding sequences tend to be better conserved than non-coding ones, as their functioning is more crucial for an organism's fitness.

Besides the conservation score, COME adds experiment-derived features as extra predictors and uses a unique decompose-compose method for feature extraction (Hu et al. 2016). In the decompose step, the reference genome is divided into bins of equal width. This allows for features to be calculated only once, server-side, providing clients with annotated genome bin data. During inference, the compose step is executed, in which transcripts aggregate the data from their corresponding bins.

2.1.1.5 Physicochemical Features

Given a transcript, we can calculate the theoretical physicochemical characteristics of the predicted peptide sequence and utilize these as features for distinguishing pcRNA from lncRNA. The isoelectric point (pI) is a frequently appearing feature in lncRNA classifiers, e.g. used by PORTRAIT, CPC2, and CPPred (Arrial et al. 2009; Y.-J. Kang et al. 2017; Tong et al. 2019), where the hypothetical pI is found to be higher for ncRNAs than for pcRNAs. Other examples of peptide-based physicochemical features include solvent accessibility (J. Liu et al. 2006) and predicted hydropathy (Arrial et al. 2009).

To be independent of the accuracy of the predicted peptide sequence (which can be limited due to the ORF finding algorithm), LncFinder directly operates on the sequence of EIIP values for a given transcript (using a known mapping) (Han, Liang, Ma, et al. 2018), extracting physicochemical features from this sequence of numbers. To do so, LncFinder first transforms the EIIP sequence into a power spectrum with FFT. A pcRNA contains sequences of nucleotide triplets (codons), hence its power spectrum has a notable peak at a third of its length. This peak is generally not observed in lncRNAs, and is therefore highly suitable as a predictive feature. On top of that, LncFinder calculates several power spectrum statistics like the signal-to-noise ratio and quantiles, using those as additional features.

2.1.1.6 Secondary Structure

Several methods have investigated the use of secondary structure information of transcripts for their classification as protein- or non-coding. Even the early classification algorithm CONC included secondary structure features as predicted by the PROFsec algorithm, although these features were part of the less contributing ones in their importance analysis (J. Liu et al. 2006).

LncFinder's feature selection procedure identified multiple secondary structure-related features to be relevant for lncRNA classification (Han, Liang, Ma, et al. 2018). LncFinder and lncRNA-MFDL use a program called RNAfold, as part of the ViennaRNA package, to find the secondary structure of a transcript based on minimum free energy. This results in a sequence of bases that are either paired (P) or unpaired (U) to a base elsewhere in the transcript (e.g. creating hairpin structures). LncFinder explores several features derived from this secondary structure sequence and uses the minimum free energy, UP frequency, and two sequence pattern distance measures in their final model. Free energy is also used by RNAplonc (Negri et al. 2018).

2.1.2 Data

Another important aspect of lncRNA classifiers is the data that is used for training and evaluation. Commonly used public databases are RefSeq (NCBI), GENCODE/Ensembl (EMBL-EBI), and NONCODE, few studies use in-house datasets. The composition of these databases is explained in Section 3.1.

RefSeq, Ensembl, and GENCODE combine automated annotation with manual curation (Frankish et al. 2022; O'Leary et al. 2015). For RefSeq, manually curated sequences are marked as 'VALIDATED' or 'REVIEWED'. The manual curation of the GENCODE and Ensembl databases is performed by the 'HAVANA' group, with reviewed sequences labeled like that. GENCODE contains only human and

mouse data, whereas Ensembl also contains other species. Nevertheless, their human/mouse annotation is exactly the same. Ensembl is therefore omitted from this study.

Finally, NONCODE is an integrated knowledge database of non-coding RNAs, mostly lncRNAs, for 17 species (L. Zhao et al. 2020). Data is obtained from public databases and literature through an automatic pipeiline and contains an advanced annotation, including e.g. expression profiles.

The majority of lncRNA classification methods undersample the majority class (pcRNA) to obtain a balanced dataset, as machine learning methods are often sensitive to class imbalance. Random sampling is a straightforward way to obtain a balanced dataset. Some previous works have aimed to handle this problem with extra care, maximizing data diversity. For example, PredLnc-GFStack and LncCat use clustering algorithm CD-HIT for this purpose (Feng et al. 2023; S. Liu et al. 2019). Alternatively, PlncRNA-HDeep clusters data based on k-mer frequencies. Another notable data manipulation techniques is applied by mRNN, which introduces mutations to training sequences as to increase the model's robustness (Hill et al. 2018).

2.1.3 Models and Methods

The third and final aspect that distinguishes an lncRNA classifier is the underlying model type, sometimes in combination with a feature selection method. Most classification tools use a machine learning or a deep learning method, but there are some that do not fall in either of these categories. For example, PhyloCSF calculates two phylogenetic trees based on a multiple sequence alignment: one under a protein-coding model, one under a non-coding model (Lin et al. 2011). The log ratio of the two phylogenetic likelihoods is used directly as a predictor for coding potential.

2.1.3.1 Machine Learning

The early lncRNA classifiers, such as CONC, CPC, and PORTRAIT, based their algorithms on support vector machines (Arrial et al. 2009; Kong et al. 2007; J. Liu et al. 2006). SVMs learn a hyperplane that maximizes the margin between the to-be-separated classes. SVMs can learn non-linear boundary functions due to the so-called kernel trick, which maps input features into a higher dimension. This does not apply to logistic regression, which can only learn linear decision boundaries, and is used by CPAT and lncScore (L. Wang et al. 2013; J. Zhao et al. 2016).

LncRNA-ID was the first of many tree-based lncRNA classification algorithms (Achawanantakun et al. 2015), using a random forest for its predictions. A random forest is an ensemble of decision trees, each trained using a slightly different feature set. This makes random forests more robust against overfitting than single decision trees or SVMs. At every node, tree-based classifiers learn to optimally distinguish two classes by creating a split that results in the highest reduction of impurity (measured in entropy, for example). This allows them to model non-linear decision boundaries, like SVMs. Random forests and decision trees were used in many later algorithms, such as PredLnc-GFStack, FEELnc, COME, and PLncPro (Hu et al. 2016; S. Liu et al. 2019; U. Singh et al. 2017; Wucher et al. 2017). LncRNA-ID stands out as it trains each of the trees in the ensemble with a different subset of coding RNAs, thereby addressing the issue of class imbalance in pcRNA/lncRNA data.

With the increasing success of boosting algorithms, three later works have based their models on XGBoost (Guo et al. 2019; M. Li et al. 2022) or CatBoost (Feng et al. 2023). Boosting creates an ensemble of learners, with each learner being trained on a dataset where each sample is given different weight. The data weights for a new learner are determined by the previous classifier's accuracy: the less accurate a previous classifier was, the more weight is assigned. This can make boosting models very powerful, but may also make them susceptible to overfitting.

2.1.3.2 Deep Learning

In recent years, deep learning-based lncRNA classification methods have started to appear more frequently. LncRNA-MFDL was the first to use deep neural networks (X.-N. Fan and S.-W. Zhang 2015), whilst still being a feature-based classification method. Other feature-based deep learning lncRNA classifiers are LncADeep and NCResNet (C. Yang et al. 2018; S. Yang et al. 2020).

Due to their ability to learn complex representations by themselves, neural networks need not to rely on a predefined set of features. As first argued by the authors of LncRNAnet (Baek et al. 2018), training networks directly on sequence data allows them to learn novel, non-canonical signals that cannot be described using traditional features. Nevertheless, neural networks require a numeric input, therefore

RNA sequences must be encoded into a sequence of numbers before training/inference. PlncRNA-HDeep combines two encoding techniques, k-mer and one-hot encoding, for their RNN and CNN, respectively (Meng, Q. Kang, et al. 2021). While neural networks are often considered to be black boxes, it is possible to get some insight into the learned features. Along these lines, mRNN performed a sequence perturbation analysis to find out which patterns are most influential for the predicted output (Hill et al. 2018).

Besides feature-based and sequence-based deep learning lncRNA classifiers, there are also those that use a combination of pre-engineered features and raw sequence data as their input (Camargo et al. 2020; X.-N. Fan, S.-W. Zhang, et al. 2020; Feng et al. 2023). The reason for this is that it is challenging for purely sequence-based networks to live up to the high performance of feature-based methods, as pointed out by a recent benchmarking paper (D. Singh et al. 2022). There are many ways in which such multimodal neural networks can be designed. RNASamba uses a so-called IGLOO network that learns embeddings for the full RNA sequence as well as the ORF, and then combines those embeddings with traditional features such as k-mer frequencies and ORF length to make a final decision (Camargo et al. 2020). Alternatively, LncCat uses a BERT model to learn embeddings for the ORF only, and then concatenates this information to traditional nucleotide and peptide features.

2.1.3.3 Feature Selection

The risk of overfitting a machine learning model increases with the dimensionality of the feature space. Therefore, several of the prior lncRNA classification studies have applied feature selection algorithms to deduce an optimal feature subset. Longdist investigated the principal components of the feature space to determine which features explained most of the data variance (Schneider et al. 2017). DeepCPP presented mDS, which selects features for which the pcRNA/lncRNA distributions are most different from each other, using relative entropy as a distance measure (Kullback-Leibler divergence). Alternatively, PredLnc-GFStack utilized a genetic algorithm for feature selection (S. Liu et al. 2019). CPC2 and LncFinder used recursive feature elimination, in which models are trained on feature subsets that shrink in size, eliminating the least contributing feature with every step.

2.2 Nucleotide Language Models

The success of BERT and other LLMs in various domains (e.g. AlphaFold in proteomics, Jumper et al. 2021) initiated the development of nucleotide language models (NLMs), pre-training on genomic data instead of natural language. Multiple of such DNA/RNA foundation models have been proposed over the past four years, and have proven to be capable of various downstream tasks such as promotor identification, variant prioritization, splice site detection, and RNA secondary structure prediction.

Table 2.3 provides an overview of existing DNA and RNA foundation models, which are further explained in this section. Both NLM types have progressed significantly, training on increasingly large datasets and incorporating the latest architectural advancements to increase efficiency and accepted input length. NLMs have also improved in terms of the applied sequence encoding method, which determines the definition of input tokens. NLMs require efficient tokenization methods to accommodate long sequences in their limited context window, causing the shift from Nucleotide-Level Tokenization (NUC) to K-mer Tokenization to Byte Pair Encoding (BPE). However, we have yet to identify a sequence encoding method that divides DNA/RNA in optimal linguistic units, especially for long sequences, which is addressed in this thesis with Research Question 4.

Note that many of the studies that are discussed here remain in pre-print, allowing them to be validated by the community and improve upon themselves through updated versions of the same work. Novel (versions of) NLMs are released every year, which indicates that this field is developing rapidly and has not converged to an optimal solution yet.

Noteworthy mentions of NLM-like models excluded from Table 2.3 are Enformer (Avsec et al. 2021), RNA-MSM (Yikun Zhang et al. 2023), MycoAI (Romeijn, Bernatavicius, et al. 2024), and DNABERT-S (Z. Zhou, Wu, et al. 2024). Reason for their exclusion is that they are released as task-specific models instead of general-purpose models, or in the case of RNA-MSM operate on multiple sequence alignments instead of single sequences.

[1]Estimated based on architecture hyperparameters, exact amount not provided in Akiyama et al. 2022

	Method	#Params	Context length	EM	Data	Reference
DNA	DNABERT-1	89M	512 (0.5 kbp)	K-mer	H	Ji et al. 2021
	DNABERT-2	117M	Variable (ALiBi)	BPE	H M	Z. Zhou, Ji, et al. 2023
	GENA-LM	110M-336M	512 (4.5 kbp) 4096 (36 kbp) Variable (RMT)	BPE	H*/M	Fishman et al. 2023
	GROVER	86M	510 (2 kbp)	BPE	H	Sanabria et al. 2024
	HyenaDNA	0.44-6.6M	64 kbp - 1 Mbp	NUC	H	Nguyen et al. 2023
	LOGO	1M	2000 (2 kbp)	K-mer	H	M. Yang et al. 2022
	NT-v1	486M-2,547M	1,024 (6 kbp)	K-mer	H*/M	Dalla-Torre et al. 2023
	NT-v2	54M-496M	2,048 (12 kbp)			
RNA	BiRNA-BERT	117M	Variable (ALiBi)	Adaptive	M	Tahmid et al. 2024
	ERNIE-RNA	86M	1024 (1 kbp)	NUC	M	Yin et al. 2024
	RiNALMo	650M	1024 (1 kb)	NUC	M	Penić et al. 2024
	RNABERT	<10M[1]	440 (0.4 kbp)	NUC	H	Akiyama et al. 2022
	RNAErnie	105M	512 (0.5 kbp)	NUC	M	N. Wang et al. 2024
	RNA-FM	100M	1024 (1 kbp)	NUC	M	Chen et al. 2022
	Uni-RNA	25M-400M	4096 (4 kbp)	NUC	M	X. Wang et al. 2023

Table 2.3: Overview of DNA/RNA Nucleotide Language Models (NLMs). Context length indicated in tokens (and bp, if applicable). EM refers to encoding method. In data column: H indicates human genome, H* indicates multiple human genomes, M indicates multi-species. Alphabetically sorted per molecule type.

2.2.1 DNA Language Models

In a pioneering study for NLMs, Ji et al. 2021 pre-trained the BERT-base architecture from Devlin et al. 2018 on sequences of 10-510 bp from a single human genome. Their model, DNABERT, has 89M parameters and outperformed state-of-the-art methods in the identification of promotor regions, splice sites, and transcription factor binding sites (at the the time of publication). Nucleotide sequences are converted into tokens through overlapping K-mer Tokenization: all k consecutive nucleotides in the sequence are considered as single tokens. Masked Language Modeling (MLM) is used as pre-training procedure, masking out k contiguous tokens in order to prevent information leakage by surrounding k-mers. The limited context length of 512 bp is addressed with the modified DNABERT-XL, which concatenates the embeddings of 512 bp subsequences. While this enables DNABERT-XL to handle longer sequences, it is still blind to long-range interactions.

A year later, a 1M parameter method named LOGO (M. Yang et al. 2022) proved that even light-weight NLMs can achieve state-of-the-art performance on tasks such as chromatin feature prediction, while being much more parameter efficient than methods like DNABERT and the CNN-based DeepSEA (J. Zhou et al. 2015). Unfortunately, LOGO was not directly compared to DNABERT or other NLMs in future works. Due to a smaller model size, LOGO could be trained on longer DNA sequences (up to 2 kbp) than DNABERT.

The Nucleotide Transformer (NT) took a much more extensive approach, training large models of up to 2.5B parameters on data from 3,202 human genomes (from the 1000 Genomes Project) and 850 multi-species genomes (Dalla-Torre et al. 2023). In their work, the authors show that increasing model size and including more (diverse) data leads to better performance on downstream tasks. The multi-species NT model performed well at prioritizing genetic variants without any fine-tuning, as it was discovered that the impact of mutations could be estimated by calculating the distance between embeddings of the sequence with and without the mutation. Finally, note that NT was trained on 12 times longer sequences (up to 6 kbp) than DNABERT due to the use of non-overlapping K-mer Tokenization (with $k = 6$) and a 2 times longer context length (1024 vs 512).

DNABERT-2 outperforms its predecessor and achieves results comparable to the 21 times larger NT-v1 with the help of an alternative sequence encoding method, multi-species pre-training, and multiple architectural improvements (Z. Zhou, Ji, et al. 2023). The authors propose the use of Byte Pair Encoding (BPE) for tokenization, a technique originally designed for data compression, which was already widely used in NLP. BPE is more efficient and less sensitive to frameshifts than K-mer Tokenization, allowing NLMs to achieve similar results with smaller models. DNABERT-2 also adapted flash attention and

low-rank adaptation to limit computational expenses, as well as Attention with Linear Biases (ALiBi) to accommodate for longer sequences. Even though ALiBi enables inference and fine-tuning on longer DNA sequences, DNABERT-2 was pre-trained on sequences of only 700 bp long. Nevertheless, DNABERT-2 is shown to outperform DNABERT-1 and NT-v1 on two downstream tasks that involve long sequences: predicting enhancer-promotor interaction and species classification.

Despite its advancements, DNABERT-2 was quickly surpassed in performance by HyenaDNA (Nguyen et al. 2023) and an updated Nucleotide Transformer, NT-v2 (Dalla-Torre et al. 2023). HyenaDNA achieved impressive results while being trained only on the human genome and with even fewer parameters than DNABERT-2 (6.6M vs 117M), which can be attributed to the use of the novel Hyena architecture (Nguyen et al. 2023). A Hyena model performs long, gated convolutions, mimicking the attention mechanism in a highly efficient manner. This enabled HyenaDNA to process sequences of up to 1 Mbp in length at nucleotide resolution, making it stand out in ultra-long range genomics tasks such as biotype and species classification. However, it was outperformed by the updated NT-v2 on multiple shorter-range tasks like chromatin profile prediction and splice site recognition. The advancements of NT-v2 in comparison to v1 include the use of rotary position embeddings, SwiGLU activation, and flash attention. This also enabled Dalla-Torre et al. 2023 to train for more epochs, on a two times longer context length (2048 tokens).

The most recent DNA language model, GENA-LM (Fishman et al. 2023) has a base context length of 36 kbp mainly due to 1) BPE tokenization with a large vocabulary size; and 2) using a sparse attention mechanism called BigBird (Zaheer et al. 2020). GENA-LM uses a BPE vocabulary size of 32,000 tokens, leading to a median token length of 9. This significantly reduces sequence length as well as model resolution. While such a large vocabulary size may introduce sampling efficiency problems (Sanabria et al. 2024), the model achieves similar performance to NT-v2 while being smaller in size (336M vs 2.5B parameters). Furthermore, a long-range GENA-LM model, which uses the Recurrent Memory Transformer technique, outperforms HyenaDNA in its own species classification benchmark. Note that GENA-LM, like NT and unlike DNABERT and HyenaDNA, was trained on data that included multiple human genomes, likely attributing to its superiority.

2.2.2 RNA Language Models

NLMs that specialize in RNA molecules have gone through similar development trajectories as DNA language models, although the amount of cross-references between the two is limited. We note that DNA and RNA LMs have not yet been thoroughly compared in literature, which is a missed opportunity since DNA language models might perform well on RNA data (indicated in Section 4.5). Regardless of that, NLMs for RNA distinguish themselves from their DNA counterparts as RNA is a single-stranded, 3D molecule that has a defined beginning and end. Furthermore, RNA data is limited to the transcriptome, excluding non-transcribed DNA regions. For these reasons, the RNA models discussed below are expected to excel at RNA-specific tasks such as secondary structure prediction and splice site detection.

The first RNA NLM, RNABERT (Akiyama et al. 2022), used a relatively small BERT architecture and pre-trained on 76,237 short, human, non-coding RNA sequences from RNAcentral. The considerably larger RNA-FM (100M parameters) was published soon thereafter (Chen et al. 2022). RNA-FM was trained on the entire multi-species RNAcentral dataset, unlike RNABERT, and accepted longer sequences than its predecessor due to a larger context length (1024 vs 440 bp). Note that both models use Nucleotide-Level Tokenization (NUC). RNA-FM was shown to be capable of modeling RNA-protein interactions and outperformed 12 state-of-the-art methods for RNA structure prediction.

The same transformer-related developments that helped improve DNA language models also contributed to better RNA language models. By utilizing rotary position embeddings and flash attention, Uni-RNA (25M-400M parameters) increased the accepted sequence length to 4096 bases (X. Wang et al. 2023). The model improved upon RNA-FM for secondary structure prediction by a large margin (F1-score of 0.82 vs 0.69), and performed well at other fine-tuning tasks such as splice site prediction and ncRNA functional classification. Unfortunately, Uni-RNA is not publically available. The RiNALMo method (650M parameters), which applies roughly the same modifications to the transformer architecture (+ SwiGLU activations), outperformed Uni-RNA on multiple downstream tasks and is available from GitHub (Penić et al. 2024). Like Uni-RNA, RiNALMo was trained on the RNAcentral dataset, but RiNALMo clustered the sequences using MMSeqs2 before pre-training, in order to maximize sequence diversity. One downside in comparison to Uni-RNA is that RiNALMo has a four times smaller context length of 1024 bp.

Recently, two RNA language models based on the Enhanced Representation with Informative Entities

(ERNIE) framework were published, each integrating a different type of knowledge into the method. ERNIE-RNA uses a special attention calculation in which pairwise position bias based on RNA structural information is integrated (Yin et al. 2024), the use of which was proven in an ablation study. The ERNIE-RNA model achieved state-of-the-art performance on multiple downstream tasks, although a thorough comparison with other RNA language models is not provided in the work. The benefit of knowledge integration was further underlined by RNAErnie (N. Wang et al. 2024), a model that takes RNA type information as extra input. Furthermore, the method applies a clever multi-level masking strategy during pre-training, in which it masks out a mix of individual nucleotides, subsequences, and database-extracted motifs. RNAErnie was found to outperform RNA-FM on tasks like RNA sequence classification, RNA-RNA interaction, and secondary structure prediction. However, a comparison to RiNALMo is not included. Even though using ERNIE was shown to improve upon a BERT-based approach, we argue that ERNIE is not suitable for the problem of lncRNA classification, as there is no additional knowledge/annotation that can be integrated into the model.

The latest model, BiRNA-BERT, addresses an issue that all previous methods shared: a limited context length (Tahmid et al. 2024). To mitigate this problem, BiRNA-BERT utilizes BPE and ALiBi, two techniques that we know from DNABERT-2 (Z. Zhou, Ji, et al. 2023). However, a unique advancement of BiRNA-BERT is that it was trained on both NUC-encoded and BPE-encoded data, such that it can handle tokens from either of the encoding methods. This way, fine-tuning tasks that involve long sequences can use the more efficient BPE-encoding method, while nucleotide-level tasks can also be approached with the same pre-trained model. In their work, the authors show that this dual tokenization scheme does not compromise the model's learning capability. Furthermore, BiRNA-BERT reaches a performance similar to that of RiNALMo, even though the latter is six times larger (650M vs 117 M parameters) and was trained for more epochs.

Chapter 3

Methods

We train lncRNA-BERT (Long Non-Coding RNA Bidirectional Encoder Representations from Transfomers), a Nucleotide Language Model (NLM) for classifying RNA as coding or long non-coding. We evaluate this method in comparison to existing lncRNA classifiers and NLMs (Research Question 1) as well as a solely feature-based approach (Research Question 2). We compare two pre-training datasets and four encoding methods to address Research Question 3 and 4, respectively. Figure 3.1 provides an overview of our method.

A specification of the utilized datasets is given in Section 3.1. Section 3.2 explains our feature-based approach to analyzing the data and performing lncRNA classification. The different encoding methods for the sequence-based NLM approach, including the novel Convolutional Sequence Encoding (CSE) method, are described in Section 3.3. Other components of our method, involving the neural architecture and training procedure, are explained in Section 3.4 and 3.5, respectively. The experimental setup is detailed in Section 3.6.

The described methodology is implemented in the Python package lncRNA-Py, which is available from GitHub (`https://github.com/luukromeijn/lncRNA-Py`) and documented on `https://luukromeijn.github.io/lncRNA-Py/`. Commands related to specific sections or results are specified throughout the text for the sake of replicability.

3.1 Data

RNA data from GENCODE, NONCODE, RefSeq, RNAcentral, and two lncRNA classification studies is used. Table 3.1 provides an overview of these data sources, Figure 3.2 shows their sequence length distributions. Section 3.1.1 specifies for every data source how the data is retrieved. For each task or experiment, we base the choice of dataset on the nature of the task, the size and reliability of the data, test set independence, and time/resource management. We distinguish between datasets used for pre-training, fine-tuning, and evaluation, which are described in Section 3.1.2.

The majority of our models are trained on human RNA data, we find that using the cross-species RNAcentral dataset for pre-training leads to a reduced performance in the downstream lncRNA classification task (Section 4.3). Moreover, from a medical perspective, we are generally more interested in human RNA than that of other species. Additional motivations for only utilizing human RNA are that these datasets 1) are believed to have higher quality annotations than other organisms; 2) contain the largest variety of (lnc)RNA sequences; and 3) are manageable in size.

3.1.1 Data Retrieval

Human pcRNAs and lncRNAs are collected from GENCODE (v46) (Frankish et al. 2022) through `https://www.gencodegenes.org/human/release_46.html`. We retrieve human non-coding RNAs from NONCODE (v6) (L. Zhao et al. 2020) via `http://v6.noncode.org/download.php`. Human RNA sequences are extracted from RefSeq (release 225) (O'Leary et al. 2015) through `https://ftp.ncbi.nlm.nih.gov/refseq/H_sapiens/mRNA_Prot/`, we then filter for 'mRNA' and 'long non-coding RNA' to isolate the pcRNAs and lncRNAs. RNA sequences from the cross-species RNAcentral (v24) (Sweeney et al. 2020) database are obtained from their FTP archive `https://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/sequences/`. Sequences with fewer than 100 nucleotides are removed

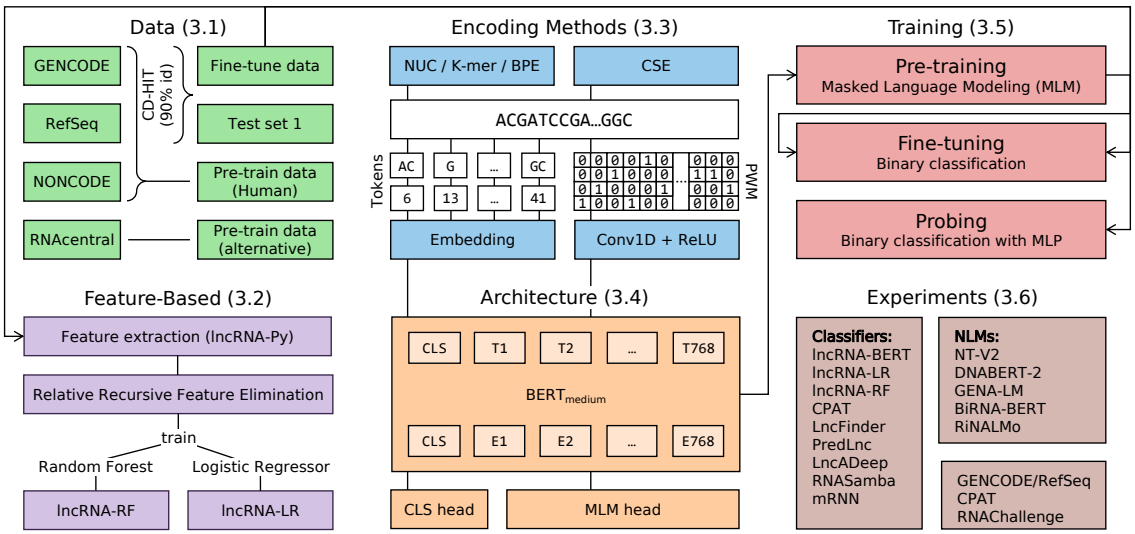

Figure 3.1: Methods overview. We pre-train lncRNA-BERT on data from GENCODE, RefSeq, and NONCODE (or alternatively, RNAcentral) and fine-tune it for lncRNA classification. Four encoding methods are compared. LncRNA-BERT is compared to a feature-based approach as well as to existing classifiers and Nucleotide Language Models. The corresponding subsections, indicated in parentheses, provide additional information.

from all of the aforementioned datasets because of our interest in long RNA molecules and to guarantee training stability.

We use two publicly available benchmarking datasets in our evaluation. The test set from CPAT (L. Wang et al. 2013) is publicly available on SourceForge (`https://sourceforge.net/projects/rna-cpat/files/test_files/`). The RNAChallenge set contains 27,283 RNA sequences which were found to be hard to classify by 48 different classification models (obtaining a maximum F1-score of 0.46) (D. Singh et al. 2022). This cross-species dataset is downloaded from `https://github.com/cbl-nabi/RNAChallenge`.

3.1.2 Definition of (Pre-)Train, Validation, and Test Sets

An overview of datasets utilized in different tasks is presented in Table 3.1. The main pre-training dataset is a human set of 297,724 coding and 238,470 non-coding RNA sequences from GENCODE (v46) Frankish et al. 2022, NONCODE (v6) L. Zhao et al. 2020, and RefSeq (v255) O'Leary et al. 2015. A randomly selected 5% of GENCODE sequences is held out for validation. The NONCODE and RefSeq datasets are used in their entirety as to maximize the number of pre-training samples. We experiment

Task	Name	# pcRNA	# ncRNA	Origin	Source
Pre-train	Human	297,724	238,470	Human	GENCODE, RefSeq, NONCODE
Validation		5,583	2,998		
Pre-train	RNAcentral	0	37,942,367	Cross-species	RNAcentral
Validation		0	2,500		
Fine-tune	GENCODE / RefSeq	101,270	48,785	Human	CD-HIT (90% id.)[a]: GENCODE, RefSeq
Validation		5,650	2,686		
Test		5,634	2,703		
	CPAT	4,000	4,000		L. Wang et al. 2013
	RNAChallenge	16,243	11,040	Cross-species	D. Singh et al. 2022

Table 3.1: Overview of the utilized datasets and the number of protein-/non-coding RNAs they contain. Section 3.1.1 describes how the data is retrieved, Section 3.1.2 explains what the different resources were used for and why. [a]The three GENCODE/RefSeq datasets are obtained by first clustering the data with CD-HIT (90% identity threshold), and then randomly selecting 90%, 5%, and 5% for fine-tuning, validation, and testing, respectively.

with RNAcentral (v24) Sweeney et al. 2020, containing 37 million ncRNAs, as alternative multi-species pre-training data source and keep aside 2,500 sequences for validating this model.

For fine-tuning the model to perform lncRNA classification, we use the CD-HIT algorithm Fu et al. 2012 to ensure non-redundancy and test set independence, similar to Feng et al. 2023; S. Liu et al. 2019. This addresses the problem of overlap between train and test sets, which is pointed out in a large lncRNA classification benchmark study (D. Singh et al. 2022). We run the CD-HIT algorithm with a 90% sequence identity threshold on the combined human pcRNA/lncRNA data from GENCODE and RefSeq, and randomly select 90% of representative sequences for training, 5% for validation, and 5% for testing. The obtained fine-tuning set contains 101,270 protein-coding and 48,785 non-coding RNAs. NONCODE was deliberately excluded during fine-tuning as to maximize data reliability.

Three test sets are used to assess the performance of lncRNA-BERT and other classifiers: GENCODE/RefSeq, CPAT, and RNAChallenge. The GENCODE/RefSeq test set (5,650 pcRNAs, 2,686 lncRNAs) is guaranteed not to overlap with our fine-tuning data because of the above-described redundancy removal with CD-HIT. The CPAT set (4,000 pcRNAs, 4,000 lncRNAs) is a published and widely used test setL. Wang et al. 2013, although some overlapping with training sets of each of the evaluated classifiers is expected. Finally, we use another published benchmark, RNAChallenge D. Singh et al. 2022, containing 27,283 hard-to-classify RNA sequences, to assess the generalizability of our model to ambiguous RNA sequences from animal, plant, and fungi species.

```
python -m experiments.create_train_test_sets
```

3.2 Feature-Based Approach

Many coding potential predictory features from previous lncRNA classifiers (Section 2.1.1) are implemented in the lncRNA-Py package, resulting in 35 feature extractor classes as well as 13 end-to-end algorithm re-implementations. Having access to a library of traditional features allows us to address Research Question 2, serving two different purposes: 1) carrying out a feature-based Exploratory Data Analysis (EDA); and 2) fitting machine learning algorithms on an optimal feature set and using these as baselines for our lncRNA-BERT model. Hereto, we extract a total of 8610 features (listed in Appendix A, Table A.3) from the GENCODE training dataset with the `thesis.feature_extraction` script.

We define two feature-based baselines of different model complexities: lncRNA-LR and lncRNA-RF, based on Logistic Regression and a Random Forest, respectively. LncRNA-LR is the simplest model trained in this work as it utilizes only 10 features, resulting in 11 trainable parameters (10 coefficients + bias). Logistic regression can be quite effective for this problem (Section 4.2, L. Wang et al. 2013) but is not capable of modeling multivariate or non-linear relationships. A random forest can model more complex decision boundaries and handles a higher number of input features with less overfitting due to its

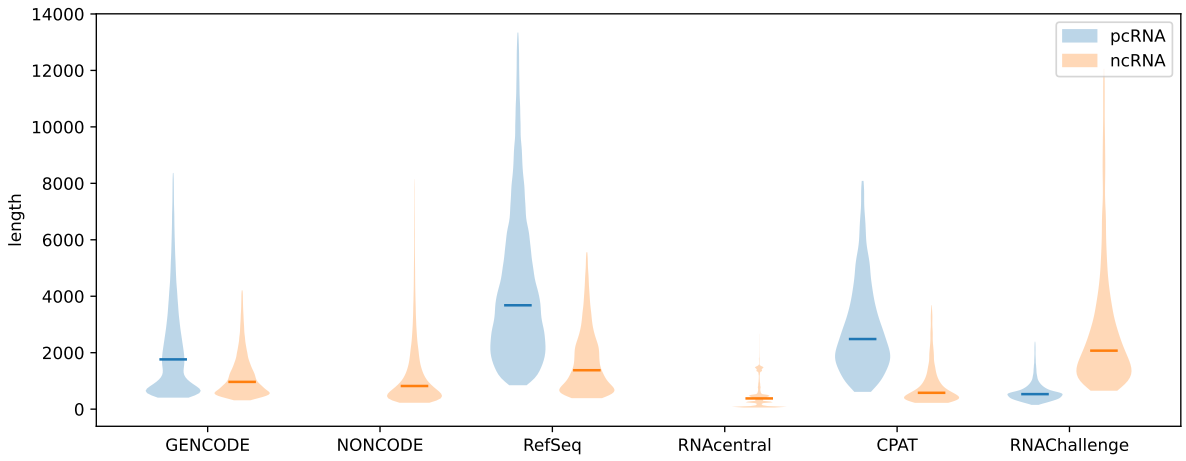

Figure 3.2: The distribution of sequence lengths per data resource, medians are highlighted. On average, pcRNAs are longer than ncRNAs, except for the RNAChallenge dataset (which likely contributes to why these RNAs are hard to classify).

ensemble-based bootstrap approach. LncRNA-RF bases its predictions on 100 features. Implementations from scikit-learn are used for both baselines, using class weights that are inversely proportional to the class size to compensate the data imbalance.

A modified Recursive Feature Eliminiation (RFE) algorithm is applied to select the most informative feature sets for lncRNA-LR and -RF (Algorithm 1), iteratively removing the 25% least important features from the feature set after fitting the model to the current set of features. Feature importance is expressed in terms of absolute coefficient size for LR and Gini importance for RF, where the Gini importance refers to the impurity reduction caused by the feature in question. Our RFE approach removes a relative (instead of fixed) number of features from the feature set per iteration, allowing the algorithm to greedily remove features in early iterations while being more considerate in later ones.

A third intended purpose of our feature library was that it enabled lncRNA-Py to serve as a standardized comparison environment to train and evaluate different 'classical' methods under the same conditions. However, we could not fully replicate all results from previous works with lncRNA-Py (Appendix A.3) due to ambiguities in the corresponding publication and/or implementation. We therefore use the official implementations of these methods in our comparison (Section 4.1).

```
python -m experiments.fit_lncrna_ml
```

Algorithm 1 Relative Recursive Feature Elimination

Input: Input matrix (X), target vector (y), features, number of features to select (s)
Output: Fitted model, features
procedure RFE(X, y, features, s)
    model ← fit_model(X, y, features)
    importances ← get_importances(model, features)
    features ← reorder(features, argsort(importances)) ▷ Sort features by importance
    $n \leftarrow \max(\lceil 0.75 \cdot |\text{features}| \rceil, s)$ ▷ Keep 75% of features, no less than s
    if $|\text{features}| > n$ **then**
        features ← select(features, n) ▷ Recursive case: select features & repeat
        return RFE(X, y, features, s)
    else
        return model, features ▷ Base case: return
    end if
end procedure

3.3 Encoding Methods

An efficient sequence encoding method is required for a transformer-based neural network to handle long RNA sequences (Research Question 4). This requirement is imposed by the transformer's attention mechanism which is quadratic in memory complexity and can therefore only be calculated on a GPU for a limited number of input positions, usually set to 512 or 1024. The number of accepted input positions is also referred to as the model's 'context length' or 'context window'. Previous works have shown that implementing architectural changes like flash attention and ALiBi can increase the accepted context length to accommodate longer sequences (Tahmid et al. 2024; Z. Zhou, Ji, et al. 2023). K-mer Tokenization and Byte Pair Encoding have been proposed as encoding methods beyond Nucleotide-Level Tokenization (Dalla-Torre et al. 2023; Z. Zhou, Ji, et al. 2023), grouping multiple nucleotides into single, pre-defined tokens. These tokens are embedded through a linear layer into d_{model} dimensions, which the transformer takes as input.

We argue that the sequence encoding methods presented in literature so far do not result in efficiently trainable representations of nucleotide sequences as they are based on large vocabularies of completely independent tokens. The novel Convolutional Sequence Encoding (CSE) method presented in this thesis can accommodate longer RNA sequences without making changes to the standard BERT architecture while maintaining nucleotide-level resolution. This is achieved by directly embedding subsequences into high-dimensional representations by means of a convolution.

We provide an extensive comparison between CSE and each of the aforementioned encoding methods (Section 4.4), which are explained in Section 3.3.1-3.3.4. An overview is provided in Table 3.2. The effect of different encoding methods on the encoded sequence length is visualized in Figure 3.3.

Method	Explanation	Example(s)	Advantages	Disadvantages
NUC	Each nucleotide is a token (A,C,T,G).	1) AGCTGCAGCGCGGGCCGC = [0,2,1,...,1,2,1]	Allows attention at highest resolution.	No sequence length reduction makes attention computationally infeasible for long sequences.
K-mer	Each k-mer is a token.	1) AGCTGCAGCGCGGGCCGC = [638, 619, 2654] 2) _GCTGCAGCGCGGGCCGCC = [2537, 2463, 2410]	Consistent token size.	Large vocabulary (4^k), token independency, k possible reading frames.
BPE	Most occurring subsequences are tokens.	1) AGCTGCAGCGCGGGCCGC = [54, 234, 2334] 2) _GCTGCAGCGCGGGCCGCC = [98, 234, 2334]	Efficient vocabulary, more robust to frameshifts.	Inconsistent token size, token independency.
CSE	Subsequences are matched against learned motifs using convolutions.	1) AGCTGCAGCGCGGGCCGC = [[0.14,0.00], [0.25,0.63], ... [0.01,0.51]]	Highly efficient 'vocabulary', forgiving for mutations.	k possible reading frames.

Table 3.2: Overview of DNA/RNA encoding methods used in NLMs: Nucleotide-Level Tokenization (NUC), K-mer Tokenization, Byte Pair Encoding (BPE), and Convolutional Sequence Encoding (CSE).

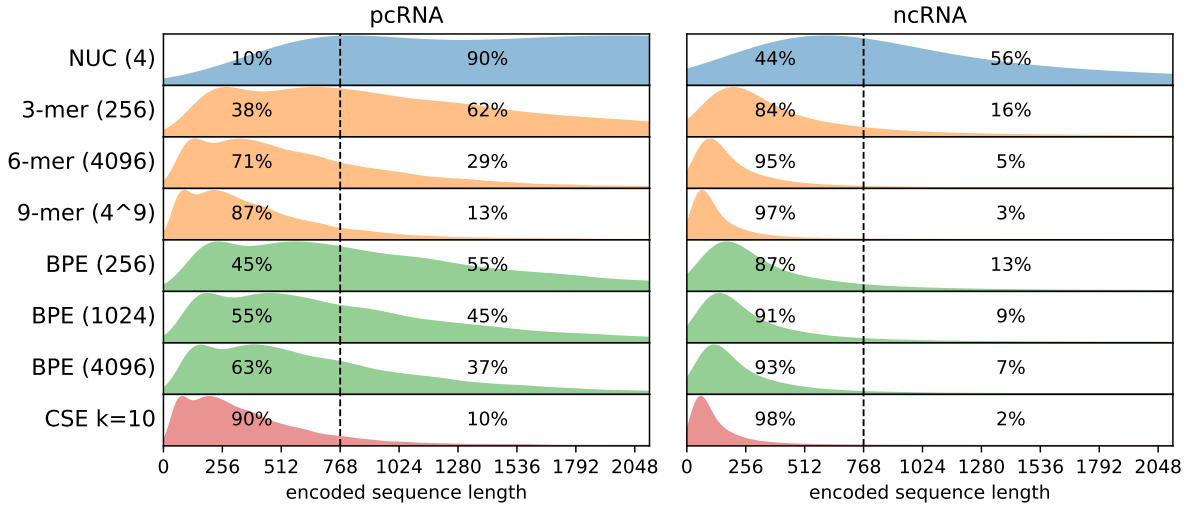

Figure 3.3: Density plots of sequence lengths from the pre-training dataset when encoded with NUC, K-mer Tokenization, BPE, and CSE. Vocabulary sizes are provided in parentheses. The percentages indicate the number of encoded sequences that fall within a medium-sized context length of 768 input positions.

3.3.1 Nucleotide-Level Tokenization

Most RNA NLMs utilize Nucleotide-Level Tokenization (NUC) as sequence encoding method (Table 2.3), having a vocabulary of four nucleotide tokens (A,C,T,G). This technique allows the transformer to calculate attention at nucleotide resolution and works well for short sequences such as the majority of RNAs in the RNAcentral dataset (Figure 3.2). However, Figure 3.3 shows that NUC is insufficient for longer RNA sequences such as pcRNAs and lncRNAs. As shown in the figure, only 44% of the non-coding RNA data in our pre-training set would fully fit into a medium-sized context length of 768 when NUC-encoded. A recent lncRNA review paper even proposed to move lncRNA's definition threshold from 200 to 500 nt, arguing that transcripts below this value are likely to correspond to different ncRNA types (Mattick et al. 2023). Hence, NUC would be insufficient for most lncRNA sequences.

3.3.2 K-mer Tokenization

K-mer Tokenization considers all possible nucleotide combinations of length k as token vocabulary and tokenizes the input as non-overlapping consecutive k-mers (Dalla-Torre et al. 2023). This reduces the sequence length by a factor of k and yields a vocabulary of size 4^k.

The method is simple and intuitive, but requires an exponentially large vocabulary size to achieve a large sequence length reduction (Figure 3.3). This introduces token sampling efficiency problems (Z. Zhou, Ji, et al. 2023) and results in an explosion of parameters in the transformer model, as each of the tokens in the vocabulary needs to have a learnable embedding. For example, for $k = 9$ and $d_{model} = 768$ we require $4^9 \times 768 \approx 201$ million parameters for the linear embedding layer alone. Besides the computational efforts that it would take to train such a model, one could argue whether all these parameters truly reflect the complexity of the data. For example, two highly similar k-mers are treated as completely independent tokens even though this may not be necessary.

Another possible limitation of this method is that it is susceptible to frameshifts. A deletion or insertion in the input causes a different k-mer reading frame and therefore a totally different sequence of tokens (Table 3.2, example 2). This means that the model has to learn k reading frames for each data signal, taking up parameters/dimensions that would preferably be dedicated to other patterns.

3.3.3 Byte Pair Encoding

The above-described downsides of K-mer Tokenization were first identified by (Z. Zhou, Ji, et al. 2023), who propose Byte Pair Encoding (BPE) as the better alternative. BPE aims to find important linguistic units by considering the most often co-occurring sets of characters (Sennrich et al. 2016). For example, the sequence 'the' is a highly common combination in English, and may be used to (partially) represent words like 'the', 'therefore', and 'thesis'. BPE is trained on an input corpus, iteratively expanding its vocabulary (initialized with the alphabet), with the most often occurring pair of subwords that are already part of the vocabulary. This process is repeated until a prespecified vocabulary size is reached. During segmentation, BPE tokenizes sequences by merging subsequences in the same order as during training.

BPE yields a fixed-size vocabulary of tokens of variable length, which gives it three advantages over K-mer Tokenization. Firstly, it can achieve a larger sequence length reduction than K-mer Tokenization with the same vocabulary size (Figure 3.3), since BPE's vocabulary contains only those k-mers that frequently occur in the data. For the same reason, BPE has a higher token sampling efficiency during pre-training. Moreover, the variable-length tokens of BPE make the method more robust against frameshifts. This is because insertions and deletions hardly affect the token merging order, causing tokenized sequences to remain largely unchanged after such events (Table 3.2, example 2).

While BPE's superiority over K-mer Tokenization has been proven multiple times in the past (Romeijn, Bernatavicius, et al. 2024; Z. Zhou, Ji, et al. 2023), it is unclear how well the BPE algorithm truly translates from application on human language to genomic language. In NLP, frequently occurring combinations of characters can be considered linguistic units, but for DNA/RNA this may not be the case. A crucial difference is that the genome has a much smaller number of characters than the human alphabet, which makes combinations of characters less rare by definition. Hence, tokens may not represent biologically informative entities, which can complicate the learning process. Also, like with K-mer Tokenization, treating all tokens as completely independent units disregards sequence similarity and may not be efficient.

Lastly, an inconsistent token length poses challenges during pre-training and fine-tuning. When using BPE with MLM, the model is tasked to predict both which and how many nucleotides appear under the mask. At the fine-tuning stage, BPE does not support nucleotide-level predictions since input positions

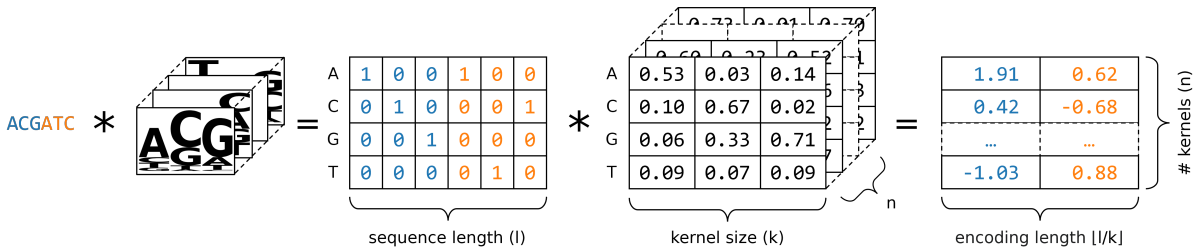

Figure 3.4: A toy example of Convolutional Sequence Encoding for sequence 'ACGATC'. Using a 1D convolutional layer with n kernels of size k and a stride of k, we can directly embed the PWM of a nucleotide sequence into an n-dimensional embedding on which a transformer can operate.

can contain a varying number of nucleotides. In contrast, K-mer embeddings can be converted back to nucleotide resolution for fine-tuning due to a fixed token size (Dalla-Torre et al. 2023).

3.3.4 Convolutional Sequence Encoding

With Convolutional Sequence Encoding (CSE, Figure 3.4), we prepend a convolutional layer to the transformer architecture to directly embed nucleotide sequences into a high-dimensional space. This way, we can effectively reduce the embedded sequence length with an efficient number of trainable parameters while maintaining nucleotide-level resolution. The idea is highly inspired by the Vision Transformer (ViT) (Dosovitskiy et al. 2020), which encodes patches of images with a small CNN before inputting them to a regular transformer architecture. A similar design has been explored before in (He et al. 2023). To the best of our knowledge, we are the first to apply this technique to accommodate long RNA sequences.

To enable a CNN to operate on an input nucleotide sequence x of length l, we can represent x as the $4 \times l$ probability distribution matrix over the four possible nucleotide bases: A, C, G, and T/U. We shall refer to this representation as x_{PWM} or the Position Weight Matrix (PWM) of x. For example, let $x =$'ACGATC', then:

$$\text{'ACGATC'}_{PWM} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

This notation is similar but not equal to one-hot encoding, the only difference being that our definition allows for IUPAC nucleotide symbols other than 'ACGT' such as 'N', which indicates equal chances for either of the four nucleotides ('N'$_{PWM} = [0.25, 0.25, 0.25, 0.25]^\top$).

A one-dimensional convolution layer with four input channels can be directly applied to PWMs of nucleotide data. Multiple previous studies have utilized this idea to enable the training of Convolutional Neural Networks (CNNs) on biological sequences (Bosco et al. 2017; Busia et al. 2018; Helaly et al. 2019). However, CNNs do not have an attention mechanism and are thus limited in their capability of modeling the long-range dependencies that are present within DNA/RNA (Ji et al. 2021).

The output of CSE can be fed directly into a transformer, which allows for attention to be calculated between each of the individual convolutions. Using n kernels of size k leads to $\lfloor l/k \rfloor$ input positions embedded in n dimensions, when the stride is also k. A transformer with $d_{model} = n$ can operate on this high-dimensional representation without requiring further embedding. An example for $k = 3$ is given in Figure 3.4. We find that activating the CSE output with ReLU increases performance (Appendix C) and add a $n \times d_{model}$ linear layer whenever $n \neq d_{model}$. Zero-padding is added to the PWM matrix to allow mini-batch training, these positions are masked out during the attention operation.

The benefits of CSE over K-mer and BPE tokenization are related to efficient parameter usage and maintaining nucleotide-level resolution. Firstly, CSE can achieve a large sequence length reduction while requiring much fewer parameters than BPE and K-mer Tokenization. E.g., the GENA-LM model uses BPE and reduces sequence length by about 9× with a vocabulary size of $32,000$ tokens (Fishman et al. 2023). This requires $32,000 \cdot d_{model}$ parameters for the embedding layer. The same length reduction with 9-mer encoding would result in $4^9 \cdot d_{model}$ parameters, while CSE can do the same with only $4 \cdot 9 \cdot d_{model}$ parameters. Thus, to increase a transformer's accepted DNA/RNA length 9 times, CSE requires approximately 888× and 600k× fewer parameters than BPE and K-mer Tokenization, respectively.

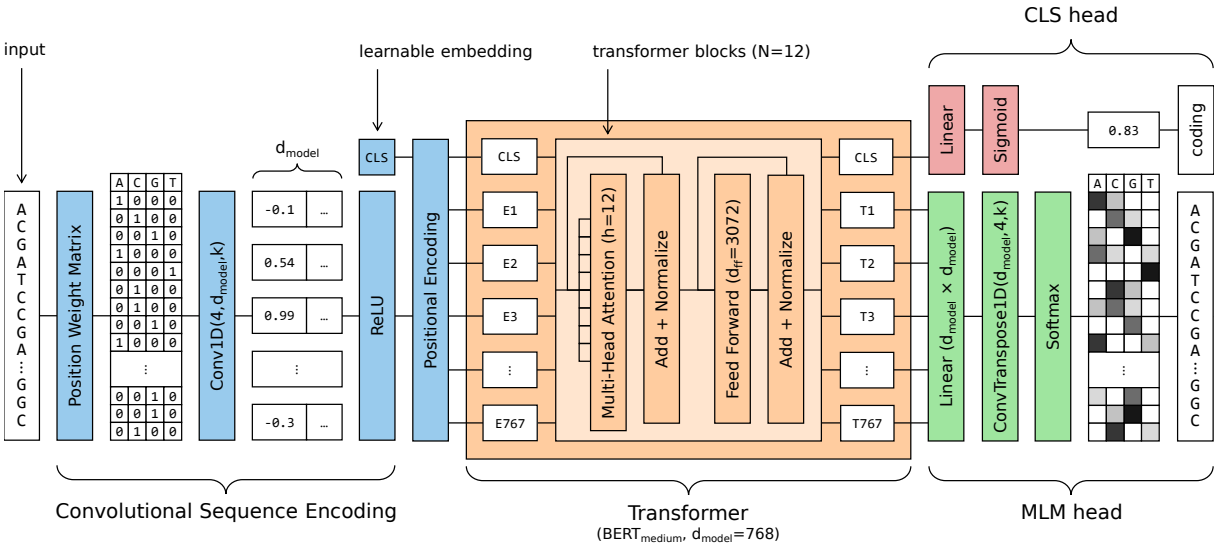

Figure 3.5: Architecture schema of lncRNA-BERT with Convolutional Sequence Encoding (CSE), which is an adaptation of the transformer encoder architecture as presented in (Vaswani et al. 2017). Fine-tuning tasks such as lncRNA classification are performed using an output head connected to the transformed `CLS` embedding. A dedicated MLM output head performs a transposed convolution, which enables masking and prediction at nucleotide resolution.

The increased efficiency and effectiveness of CSE is achieved by incorporating into its design that k-mers are combinations of nucleotides instead of completely independent units. For example, two k-mers with a single nucleotide difference will be modeled closely together with CSE by design, while a BPE or K-mer based model might need several training epochs to learn this information. Of course, a single mutation could be highly impactful for the meaning of a subsequence, but this meaning is context-dependent and cannot be fully captured in the initial embedding layer, regardless of the encoding method.

A final advantage of CSE is that it has a consistent token size, unlike BPE. This allows nucleotide-resolution training tasks, such as nucleotide level MLM during pre-training (Section 3.5.1) and fine-tuning for splice site prediction (Tahmid et al. 2024).

Downsides to the CSE approach include frameshift sensitivity, fine-tuning convergence issues, and mutation insensitivity. The frameshift sensitivity problem is similar to that of K-mer Tokenization, as sequences can be observed in k different reading frames. In Section 4.4.1, we show that this does not affect the overall embedding of a sequence, as long as $k \mod 3 \neq 0$. We attribute fine-tuning convergence issues to the BERT model's reliance on the CSE encodings, the latter of which are constantly updated during training. Finally, we anticipate that the model might be slightly insensitive to local mutations, as subsequences with mutations will be encoded in close proximity of the original.

3.4 Neural Architecture

We adapt the transformer encoder architecture from (Vaswani et al. 2017) as used by BERT (Devlin et al. 2018) with some minor adjustments to incorporate CSE (Figure 3.5). The embeddings generated by CSE are enriched with positional information by adding a fixed sinusoidal positional encoding (Vaswani et al. 2017). The `CLS` token from BERT is replaced with a learnable `CLS` embedding, like in ViT (Dosovitskiy et al. 2020), as CSE does not tokenize the input but directly embeds it. The `CLS` embedding is always inserted as the first input position. The transformed embedding of the `CLS` token is used as input to the lncRNA classification output head, which is a sigmoid-activated linear layer containing a single node.

Our experiments are performed with a model configuration like $\text{BERT}_{\text{medium}}$, with $N = 12$ transformer blocks, a dimensionality of $d_{model} = 768$, $d_{ff} = 3072$ nodes in the feed-forward layers, and $h = 12$ attention heads.

LncRNA-BERT uses a medium-sized context length of $c = 768$ input positions. Hereto, we set the zero-padded PWM input length to $k \cdot (c-1)$, such that we obtain c input positions when prepending the `CLS` token.

To perform Masked Language Modeling (MLM) with CSE, a transposed convolution layer is used to deconvolve the transformed embedding for every input position (except `CLS`) into k predictions at nucleotide level, using a stride/kernel size of k. When Softmax-activated, the output represents a probability distribution (PWM) over the four possible nucleotides for $k \cdot (c-1)$ input bases. This enables Masked Language Modeling at the nucleotide level. We find that first performing a linear transformation ($d_{model} \times n$) of the transformer output before the transposed convolution operation is beneficial for convergence (Appendix C).

3.5 Training

We pre-train lncRNA-BERT for 7 days, after identifying an optimal model configuration based on pre-training, fine-tuning, and probing. Optimal model checkpoints are stored based on validation set performance.

3.5.1 Pre-training

To pre-train our RNA model, we adapt the Masked Language Modeling (MLM) task (Devlin et al. 2018) to CSE. The convolutional input layer of our network and the transposed convolution within its MLM output head allows us to introduce masks at nucleotide resolution. Analogous to the dedicated `MASK` token used in most BERT models, we use the IUPAC symbol 'N' to mask out specific nucleotides. This character lends itself naturally for the purpose of masking as it indicates an equal chance of being either of the four canonical bases.

Like in standard MLM, a $p_{mask} = 0.8$ proportion of the selected nucleotides are masked out (using 'N'), while a $p_{random} = 0.1$ part is randomly changed. These operations are arguably more meaningful for DNA than for natural language. Uncertainly sequenced bases ('N') and random mutations are highly common in DNA data, while the `MASK` token or random replacement of words do not occur in natural language. Hence, MLM does not only pre-train our model, it also makes it more robust to sequencing errors and mutations.

The model is pre-trained for 7 days using a cross entropy loss function, a batch size of 8, and the Adam optimizer in combination with a learning rate schedule proposed in (Vaswani et al. 2017), with 32,000 warmup steps. The pre-training dataset (Section 3.1) consists of a total of 536,194 RNA samples and is seen approximately 20 times during training. An important detail is that when using CSE, a random number of up to $k-1$ bases is removed from the input, such that every sequence in the data is seen in multiple reading frames. This is a countermeasure to the reading frame sensitivity of CSE described in Section 3.3.4. In addition, when a sequence does not fully fit into CSE's context length, a random subsequence is input to the model, as in (Penić et al. 2024).

The pre-training script is called via the following command (only basic arguments listed, see `https://luukromeijn.github.io/lncRNA-Py/scripts.html` for full documentation).

```
python -m lncrnapy.scripts.pretrain \
    --encoding_method {cse,bpe,kmer,nuc}] \
    [--n_kernels N_KERNELS] \
    [--kernel_size KERNEL_SIZE] \
    [--bpe_file BPE_FILE] \
    [--k K]
    fasta_train fasta_valid
```

3.5.2 Fine-tuning

The model is fine-tuned for lncRNA classification on the fine-tuning dataset (Section 3.1) containing 100,587 coding and 53,868 long non-coding RNAs. Optimization is done for 100 epochs of 10,000 samples, using Adam, a fixed learning rate of 10^{-5}, and a batch size of 8. We use a binary cross entropy loss function with the reciprocal class sizes as weights to counteract the class imbalance. Like in MLM, the CSE-based model is randomly input with one of k possible reading frames.

```
python -m lncrnapy.scripts.train \
    [--pretrained_model PRE-TRAINED_MODEL]
    [--encoding_method {cse,bpe,kmer,nuc}]
    [--learning_rate LEARNING_RATE]
```

```
    [--bpe_file BPE_FILE]
    [--k K]
    fasta_pcrna_train fasta_ncrna_train
    fasta_pcrna_valid fasta_ncrna_valid
```

3.5.3 Probing

We report probing performance to assess the informativeness of the embeddings generated by our models without fine-tuning them, similar to (Dalla-Torre et al. 2023). Hereto, we train a small Multi-Layer Perceptron (MLP) with a single hidden layer of 256 nodes on the mean-pooled output embeddings of our models (using the `--freeze_network` and `--hidden_cls_layers` flags of `lncrnapy.scripts.train`, with an increased learning rate of 0.0001).

3.6 Experimental Setup

In our experiments, we identify an optimal configuration of lncRNA-BERT and compare it to eight alternative lncRNA classification algorithms. These include lncRNA-LR and -RF (Section 3.2), as well as six algorithms presented in previous works, three of which are based on deep learning. Algorithms are selected from Table 2.2 based on their relevance to this thesis as well as to ensure a diverse comparison set. CPAT, a feature-based logistic regression algorithm (L. Wang et al. 2013, downloaded from `https://sourceforge.net/projects/rna-cpat`) is included because it is currently integrated into LUMC's RNASeq pipeline. LncFinder is based on an SVM (Han, Liang, Ma, et al. 2018, downloaded from `https://cran.r-project.org/package=LncFinder`) and uses features that obtain a high ranking in our feature selection procedure (Section 4.2.3). PredLnc-GFStack (PredLnc) is the most complex ML-based approach included in this comparison, using an ensemble of random forests with optimal feature sets (S. Liu et al. 2019, downloaded from `https://github.com/BioMedicalBigDataMiningLab/PredLnc-GFStack`). The deep learning methods in our analysis include the feature-based LncADeep (C. Yang et al. 2018, downloaded from `https://github.com/cyang235/LncADeep`), sequence-based mRNN (Hill et al. 2018, downloaded from `https://github.com/hendrixlab/mRNN`), and feature/sequence hybrid method RNAsamba (Camargo et al. 2020, accessed through web server `https://rnasamba.lge.ibi.unicamp.br`). Both LncADeep and RNASamba were ranked among the top five lncRNA classification algorithms in a recent benchmark (D. Singh et al. 2022). Out-of-the-box models are used, without re-training.

Hyperparameter tuning (Appendix C, `https://luukromeijn.github.io/lncRNA-Py/experiments.html#hyperparameter-tuning`) as well as extensive comparisons between different encoding methods and pre-training datasets are conducted to identify an optimal configuration for lncRNA-BERT. All models in the comparisons were pre-trained and fine-tuned for 500 and 100 epochs of 10,000 samples, which takes roughly 2 and 0.5 days, respectively. During training, we store optimal model checkpoints by evaluating performance on the validation set after every training epoch, assessed via MLM accuracy during pre-training and macro-averaged F1-score during fine-tuning.

Based on our results, two lncRNA-BERT models with optimal encoding methods (3-mer tokenization and CSE with $k = 9$) and pre-training data (human mRNA/lncRNA) for lncRNA classification were pre-trained for an extra long period of 7 days (automatically terminated by workload manager) and fine-tuned for 100 epochs. Training was carried out using the compute resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by Leiden University (CPU: AMD EPYC 7513 2.6GHz, GPU: A100 MIG 4g.40GB). Other experiments were conducted on ALICE, the HPC cluster SHARK from LUMC, or HPCs from LIACS' DSlab.

Chapter 4

Results

We identify pre-training on human data and using CSE ($k = 9$) or 3-mer tokenization as encoding methods as optimal model configurations for lncRNA-BERT and demonstrate that these model achieve performance competitive with six previously proposed classifiers.

We shall discuss the results in the following order. Section 4.1 evaluates the performance of lncRNA-BERT in comparison to existing lncRNA classifiers (Research Question 1). Section 4.2 describes the feature-based baselines and highlights several results from our exploratory data analysis (Research Question 2). Section 4.3 discusses the effect of pre-training data on the behaviour of the model (Research Question 3). An in-depth comparison of alternative sequence encoding methods is given in Section 4.4 (Research Question 4). We then compare lncRNA-BERT to NLMs presented in previous works in Section 4.5 (expanding upon Research Question 1). Finally, Section 4.6 assesses the potential meaning of the embeddings learned by lncRNA-BERT (related to Research Question 3).

4.1 Comparison to Established LncRNA Classifiers

Figure 4.1 shows that our lncRNA-BERT models achieve a performance (F1 ≈ 0.94 on GENCODE/RefSeq, ≈ 0.95 CPAT) similar to the best algorithms included in our comparison, proving that using an NLM is a valid approach for the problem of lncRNA classification. However, lncRNA-BERT does not improve upon the established methods by a large margin. Interestingly, our feature-based lncRNA-RF model consistently outperforms all of the benchmarked methods on all three test sets (in terms of F1-score), including lncRNA-BERT. LncADeep obtains the most consistent and high ranking out of all of the classifiers presented in previous works, but is outperformed by lncRNA-RF.

The remaining methods show varying performances for different test sets, e.g. CPAT performs particularly well on its own test set. We attribute these differences to overlap and similarities between train and test sets, i.e. CPAT's test set is likely to be similarly distributed as the training set. This bias in our analysis is mitigated for lncRNA-BERT, -LR, and -RF by generating an independent train/test split after redundancy removal with CD-HIT (Section 3.1.2).

In general, the three test sets exhibit varying difficulty levels, with RNAChallenge being the most difficult (mean F1: 0.15), followed by GENCODE/RefSeq (mean F1: 0.91), and CPAT (mean F1: 0.96). The low scores on the RNAChallenge dataset are caused by it being a multi-species test set containing transcripts that most algorithms fail to classify correctly (D. Singh et al. 2022). A high performance is therefore unexpected, but the obtained scores are an indication of the (cross-species) generalization capabilities of a model. LncRNA-RF obtains the highest F1-score on the RNAChallenge set (0.316), followed by lncRNA-BERT with CSE (0.242) and 3-mer tokenization (0.235). The superior F1-scores of our models on RNAChallenge relate to the obtained values for precision and recall on ncRNA, which are zero for all methods in the analysis except for ours. The cause of this may be related to the GENCODE/RefSeq training set, as it is the only aspect that overlaps between our BERT, RF, and LR models.

4.2 Feature-Based Exploratory Data Analysis and Classification Baselines

The performance of our two feature-based classification baselines, lncRNA-LR (logistic regression, 10 features) and lncRNA-RF (random forest, 100 features) is shown in Figure 4.1. LncRNA-RF demonstrates

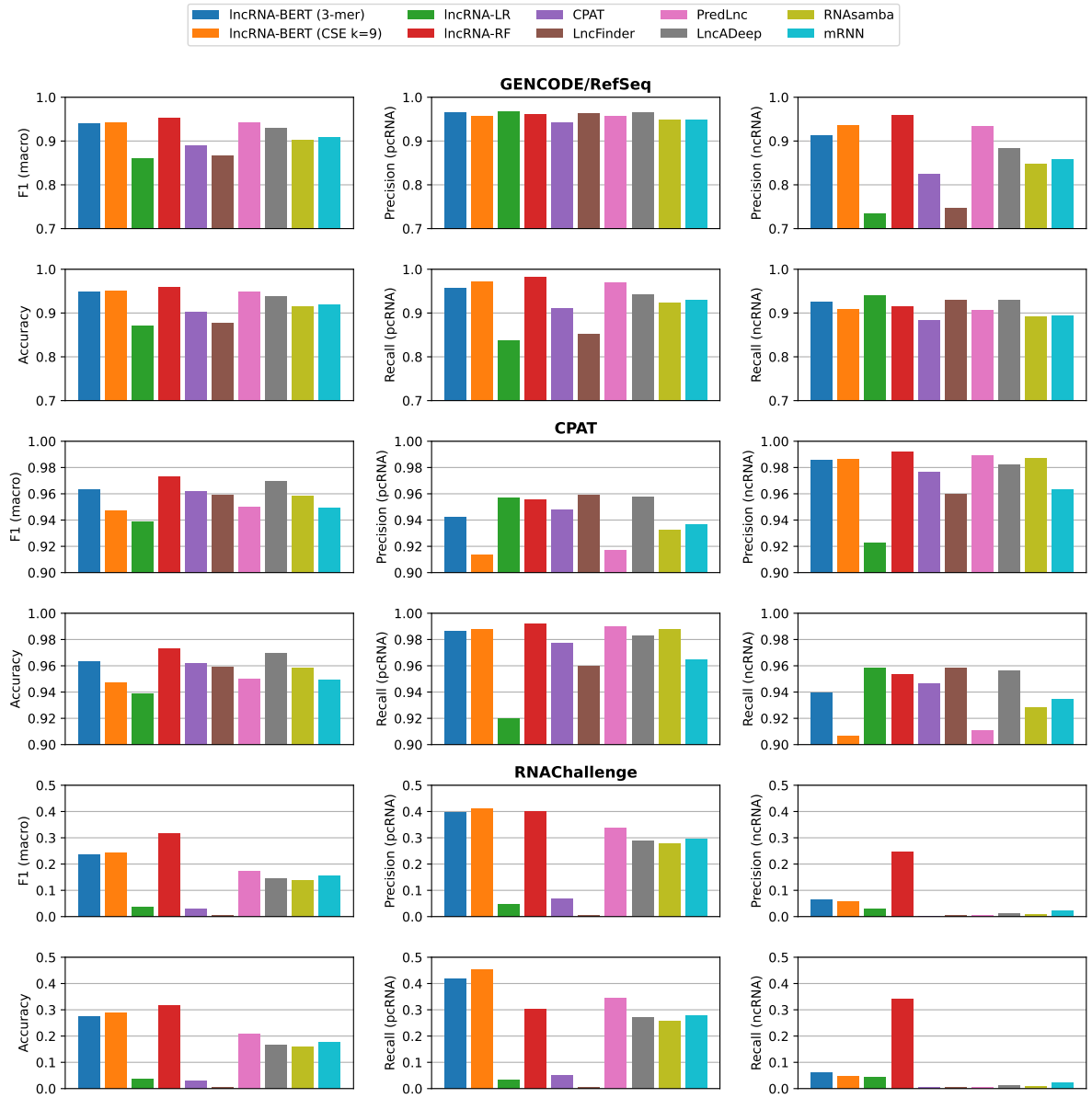

Figure 4.1: Performance of lncRNA classifiers on three test sets: 1) 5,634 pcRNAs and 2,703 lncRNAs from GENCODE/RefSeq, held out from the training sets of lncRNA-BERT, -LR, and -RF, after redundancy removal with CD-HIT; 2) test set from CPAT, containing 4,000 pcRNAs and 4,000 lncRNAs; and 3) the cross-species RNAChallenge dataset (D. Singh et al. 2022). LncRNA-BERT ranks among the best-performing classifiers. LncRNA-RF consistently outperforms the other methods. Values are listed in Appendix B, Table B.1.

Method	# Parameters	F1 (macro)		
		GENCODE/RefSeq	CPAT	RNAChallenge
lncRNA-LR	11	0.870	0.939	0.037
lncRNA-RF	6.2M	0.960	0.973	0.316
lncRNA-BERT (3-mer)	88.9M	0.940	0.963	0.235
lncRNA-BERT (CSE k=9)	88.9M	0.943	0.947	0.242

Table 4.1: Number of trainable parameters for each of our models, and their macro-averaged F1-scores on the GENCODE/RefSeq, CPAT, and RNAChallenge test sets.

an improved performance in comparison to lncRNA classifiers from previous works and also outperforms lncRNA-BERT on all three test sets. This is noteworthy, as lncRNA-RF is a purely feature-based approach and has 14× fewer trainable parameters (Table 4.1).

Optimal feature sets are selected for both classifiers using recursive RFE (Section 3.2, Algorithm 1). The 25 most important features are listed in Table A.2 (Appendix A) and mostly include ORF- and alignment-based features. This affirms the intuition that the presence of an ORF and the number of alignment hits with a protein database can be used to predict coding potential. This is similar to what has been reported in many previous works.

The Exploratory Data Analysis (EDA) as reported in this section focuses on sequence-intrinsic features that can help to identify the limitations of feature-based lncRNA classifiers and motivate the use of a NLM for this task. We first show that ORF detection alone is not sufficient to distinguish coding from non-coding transcripts. Then, we identify the importance of k-mers and the limitations of using their occurrence frequencies as coding potential predictors. Finally, we show that pcRNA and lncRNA exhibit a different high-level organization, with pcRNA having a higher entropy and a stronger three-base periodicity. The EDA is carried out on a dataset of solely GENCODE sequences, instead of the GENCODE/RefSeq training set, as the latter was defined at a later stage in the development of our method.

```
python -m experiments.eda
```

4.2.1 Most LncRNAs Contain ORFs of Limited Length

Inspecting the ORF lengths of GENCODE transcripts (Figure 4.2) affirms previous findings that lncRNAs can have (short) ORFs (Pang et al. 2018). Our most strict ORF finding algorithm looks for the longest possible subsequence of nucleotide triplets between a start and stop codon and identifies ORFs in 94% and 95% of the pcRNAs and lncRNAs in GENCODE, with a mean length of 1195 and 218 nucleotides, respectively. The presence of an ORF does not necessarily imply translation into a functional protein, as this is also dependent on other factors such as regulatory signals and the secondary structure of the RNA molecule. Nevertheless, these findings underline that ORF identification is not sufficient for distinguishing coding from non-coding transcripts.

Figure 4.2 shows that ORF coverage is an informative feature for longer RNAs, as the ORF length of lncRNAs is limited to about 1000 nt regardless of the full sequence length. In contrast, the ORF length of pcRNA increases with the length of the sequence (Pearson's $R = 0.81$ for pcRNA versus $R = 0.24$ for lncRNA).

4.2.2 PcRNA and LncRNA Differ in K-Mer Composition

Figure 4.3 shows that the occurrence of specific k-mers can be a useful variable for predicting coding potential, as also reported in previous works (A. Li et al. 2014; J. Liu et al. 2006; Wucher et al. 2017). The left panel of Figure 4.3 shows that pcRNAs and lncRNAs somewhat segregate within the 6-mer frequency space (even though in Section 4.6 we find that lncRNA-BERT can obtain more informative embeddings). We fit a random forest on 6-mer spectra of the GENCODE training set and report a macro-averaged F1-score of 86% on the validation dataset. This score demonstrates the usefulness of k-mer features, despite not being competitive with other lncRNA classifiers that use additional covariates. The score also sets a lower-bound for the performance of an NLM, in which k-mer features are enriched with positional and contextual information.

The two right panels of Figure 4.3 show density plots of the most important 3-mer ('CGA') and 6-mer ('GGCGGC') frequencies in random forests (evaluated with Gini importance) that we fit solely on those

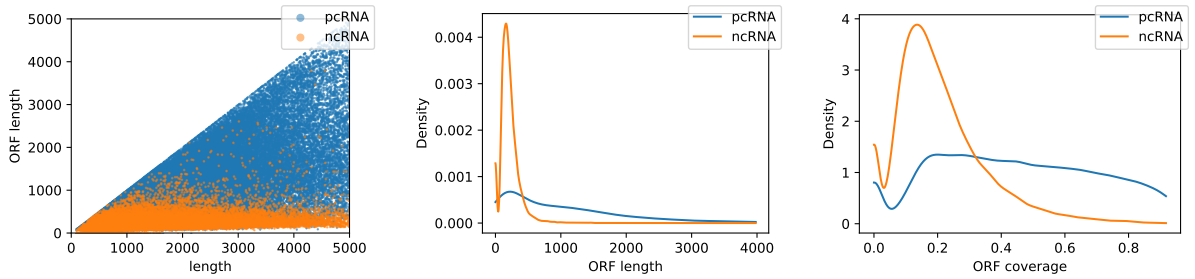

Figure 4.2: Scatter plot (left) and density plots (middle, right) of length and ORF length of pcRNA and lncRNA in GENCODE. Both pcRNA and lncRNA contain ORFs, but the ORFs in lncRNA are limited in length (< 1000 nt) compared to those in pcRNA. Consequently, the average ORF coverage is higher in pcRNA than in lncRNA.

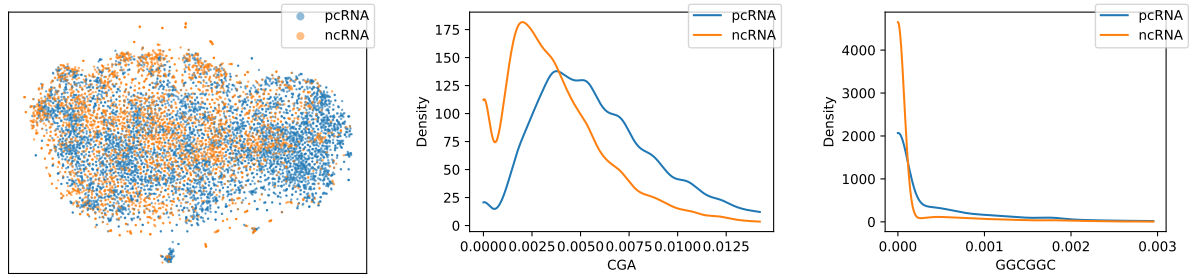

Figure 4.3: T-SNE visualization of the 6-mer frequency space of the validation dataset (left) and density plots of the most discriminative 3-mer ('CGA', middle) and 6-mer ('GGCGGC', right) frequency features of pcRNA and lncRNA in GENCODE. There are certain k-mers with a different occurrence frequency in coding versus non-coding RNA, yet these are not sufficient to fully distinguish between the two classes.

frequencies. We report a statistically significant difference between their means for pcRNA and lncRNA ($P = 0$ for both 'CGA' and 'GGCGGC') and count 63 and 3508 statistically significant associations for all 3-mers and 6-mers, respectively (using $\alpha = 0.05/4^k$) . The biological significance of these k-mers could not be verified via literature and may be explored in future work. ORF-based k-mer frequencies correspond to in-frame occurrences (when assuming a correct ORF identification) and therefore rank higher in the feature importance list of lncRNA-LR and lncRNA-RF (Appendix A, Table A.2).

4.2.3 LncRNAs Are Organized Differently than PcRNAs

We show that sequence entropy is an important variable for distinguish coding from non-coding RNA (also see Appendix A.2, Table A.2) and prove that LncFinder's k-mer distance feature is a useful metric to describe this information (Figure 4.4). The difference in ORF 3-mer frequency entropy between coding and non-coding transcripts is attributed to pcRNA having a more equally distributed k-mer spectrum than lncRNA, while lncRNA contains more copies of the same k-mers. These unequally distributed k-mer spectra may be related to repeat regions or a less complicated organization of lncRNA compared to pcRNA in general. One could suspect sequence length to be an influential factor here, but we report a correlation between length and entropy of only $R = 0.34$.

The k-mer distance was designed to describe the similarity between an input RNA and the average (long non-)coding RNA, using k-mer spectra as high-dimensional sequence representations (Han, Liang, Ma, et al. 2018). However, the high correlation between entropy and k-mer distance (Figure 4.4, left) indicates that this feature merely describes the entropy of a sequence. This is underlined by the observation that the k-mer distance of pcRNAs to the average lncRNA is lower than that of most lncRNAs (Figure 4.4, right). In other words, the metric fails to capture the idea of similarity that (Han, Liang, Ma, et al. 2018) designed it for, yet effectively models the entropy of a sequence. This makes it an important feature in our lncRNA-LR and lncRNA-RF models.

Three-base periodicity is another feature related to sequence organization that helps discriminate the two RNA classes and is of high importance in lncRNA-LR and lncRNA-RF. The biological interpretation of this feature is that most lncRNAs do not code for functional proteins and therefore do not adhere to

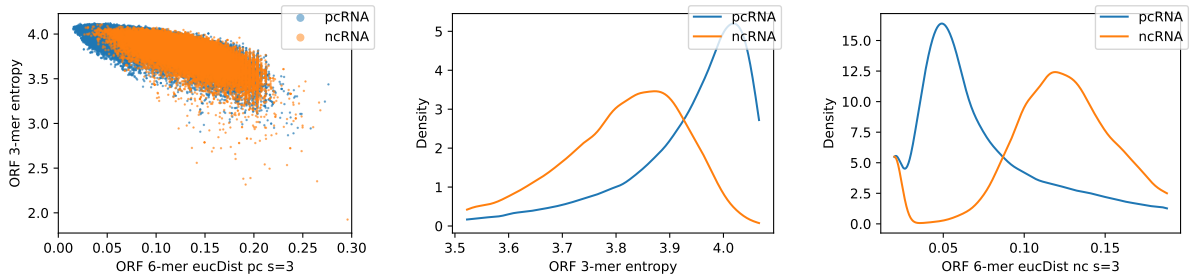

Figure 4.4: Scatter plot (left) and density plots (middle, right) of the ORF 3-mer entropy and 6-mer distance of pcRNA and lncRNA in GENCODE. Left: The two features are negatively correlated ($R = -0.82$). Middle: The ORF 3-mer frequency spectrum of pcRNA tends to have a higher entropy than that of ncRNA, indicating that coding RNA contains a higher variety of 3-mers. Right: This causes the distance to the average ncRNA to be lower for pcRNA than for ncRNA itself.

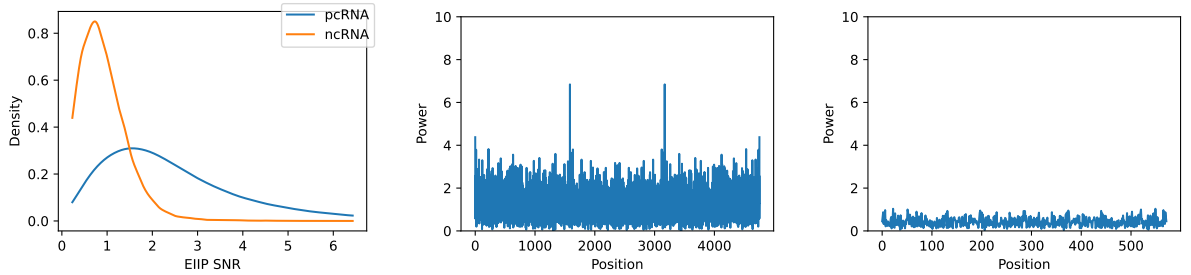

Figure 4.5: Density plot of pcRNA and lncRNA in GENCODE (left) and power spectrum plots for a randomly chosen pcRNA (middle) and lncRNA (right) of the Electron Ion Interaction Profile (EIIP) feature as used by LncFinder (Han, Liang, Ma, et al. 2018). PcRNA tends to have a peak at 1/3 position of the power spectrum due to its 3-base periodicity, leading to a higher signal-to-noise ratio than lncRNA.

the codon structure that pcRNAs exhibit. It is quantified by using the Electron Ion Interaction Potential (EIIP) spectrum, which for pcRNA usually contains a peak at 1/3 position (Han, Liang, Ma, et al. 2018). Figure 4.5 shows the density plot of the Signal-to-Noise ratio of this spectrum, as well as example spectra for a coding and non-coding transcript.

4.3 Pre-Training

This section shows the importance of pre-training and demonstrates the benefits and downsides of using the alternative multi-species RNAcentral dataset in comparison to the human dataset for pre-training.

4.3.1 Pre-Training on Human RNA Data Leads to Highest Classification Performance

Figure 4.6 shows that our models converge faster and achieve the highest lncRNA classification performance when pre-trained on human data. This confirms that pre-training helps to achieve the highest downstream performance. The figure also indicates that our human pre-training dataset is more suitable than RNAcentral for the downstream classification task. The performance difference is explained by the high similarity between the human pre-training and fine-tuning datasets, which are both comprised of only human pcRNA and lncRNA. On the other hand, the RNAcentral dataset contains all types of non-coding RNA across a wide range of species. The underrepresentation of (human) lncRNA and complete lack of pcRNA makes the RNAcentral model less familiar with the fine-tuning data (see Figure 4.7), leading to a decreased convergence rate and F1 score.

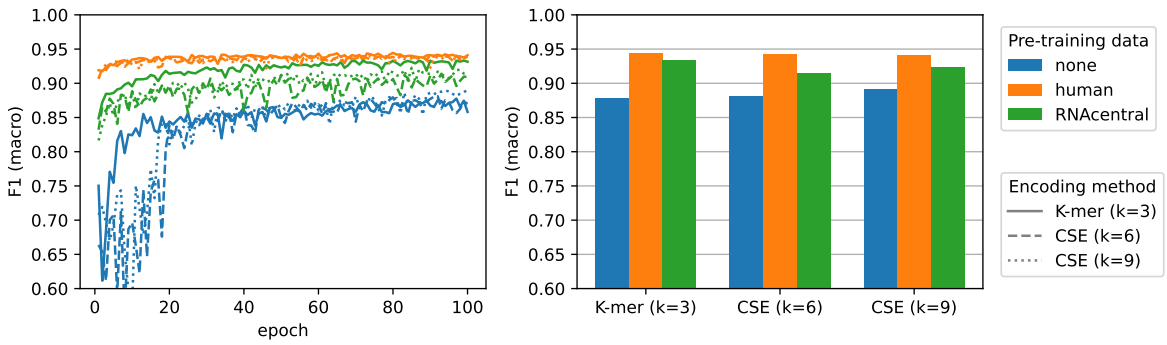

Figure 4.6: Macro-averaged F1-score on the validation set during (left) and after (right) training for 100 epochs on the lncRNA classification task, for different pre-training configurations and encoding methods. The models benefit from pre-training, leading to faster convergence and increased performance. Pre-training on human data leads to the highest downstream performance.

4.3.2 RNAcentral Pre-Training Causes Model to Prioritize Different RNA Types

The difference between pre-training on human/RNAcentral data becomes more apparent when inspecting the model's embeddings and predictions. Figure 4.7 visualizes the latent space and MLM performance of a model pre-trained on RNAcentral, in comparison to the human model (both using 3-mer tokenization).

Figure 4.7 shows that the human model has learned to distinguish pcRNA from lncRNA in its embedding space after pre-training. The emergence of such distinction is noteworthy, as our pre-training task (MLM) is a self-supervised procedure that is independent of target labels. This finding therefore indicates that coding potential is a prominent, sequence-intrinsic signal. While lncRNA-BERT is not the first method to solely base its predictions on sequence patterns (Hill et al. 2018; A. Li et al. 2014; Liang Sun et al. 2013), it is the first to be capable of discriminating between pcRNA and lncRNA in a fully self-supervised manner.

In comparison, a model pre-trained on RNAcentral generates a less convincing distinction between the two classes and is shown to specialize into different RNA types instead. RNAcentral does not contain data annotations for all of its sequences. Therefore, we inspect the model's performance on other RNA types by visualizing a labeled dataset of 3,992 RNA sequences across 10 structural families (Sloma et al. 2016, data access granted upon author's request). Here, we show that the embeddings generated by our RNAcentral model successfully separates the different families, while the human model fails to obtain a similar separation. BiRNA-BERT and RiNALMo have used the same dataset to show that their models, which use RNAcentral as main pre-training resource, can also distinguish the different families within the ArchiveII dataset (Penić et al. 2024; Tahmid et al. 2024).

Inspecting the MLM accuracy (% of correctly predicted masked nucleotides) per sequence also shows that an RNAcentral pre-trained model performs well on certain RNA sequences (mean: 80%) but does not achieve the same accuracy on human pcRNA/lncRNA data (mean: 43%). It is clear from Figure 4.7 that the model learns, but the learned patterns do not generalize to lncRNA. A higher MLM accuracy on human RNA is achieved when using our human dataset for pre-training instead of RNAcentral, even though this accuracy (mean: 18%) is far from the high values that an RNAcentral model achieves when evaluating on the RNAcentral validation set. We attribute this to the sequences in the RNAcentral dataset, which are higher in number, more evolutionary diverse, smaller in size, and lower in complexity. We also anticipate that certain RNAs are highly overrepresented in the dataset, making the model prefer those sequences over lncRNA.

The results in Figure 4.6 and 4.7 expose a potential weakness of RNA language models that have only been trained on RNAcentral data, like ERNIE-RNA, RNABERT, RNAErnie, and RNA-FM (Akiyama et al. 2022; Chen et al. 2022; N. Wang et al. 2024; Yin et al. 2024). These models are likely to underperform on lncRNA/mRNA, leading to a diminished lncRNA classification performance. It is unclear whether BiRNA-BERT, RiNALMo, and Uni-RNA suffer from the same issue. These works also utilize the RNAcentral dataset but augment it with data from RefSeq (Tahmid et al. 2024), Rfam/Ensembl (Penić et al. 2024), and Genomic Warehouse (X. Wang et al. 2023). We anticipate that a cross-species pre-training

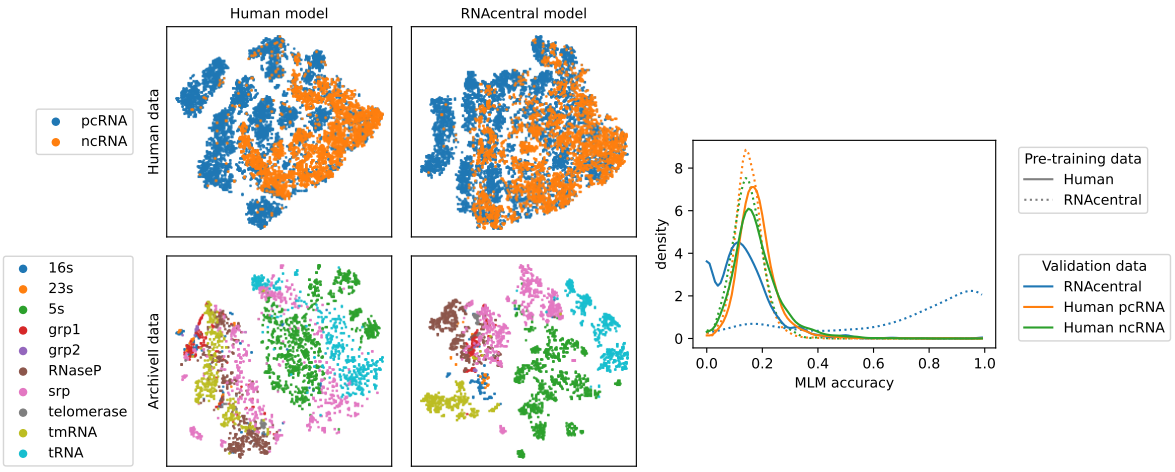

Figure 4.7: The effect of pre-training data on lncRNA-BERT. Left: T-SNE visualizations of the embedding spaces of the human validation set and the ArchiveII dataset after pre-training lncRNA-BERT with 3-mer tokenization on human data or on multi-species data from RNAcentral. The human model better distinguishes pcRNA from ncRNA than the RNAcentral model, while the latter generates an improved separation between structural families in ArchiveII. Right: Density plot of the MLM accuracy per sequence (% correctly predicted tokens) for lncRNA-BERT with 3-mer tokenization when pre-trained/evaluated on human pcRNA/ncRNA data or cross-species ncRNA from RNAcentral. The RNAcentral model achieves a high MLM accuracy (mean: 70%) on the RNAcentral validation set but performs worse than the human model on lncRNA from the human validation set (mean: 15% versus 18%). The models slightly favor pcRNA over ncRNA.

dataset could enhance performance but should be of good quality, well-balanced, non-redundant, and include RNA of all types.

4.4 Encoding Methods

We compare nucleotide encoding methods (Section 3.3) and show that 1) our novel CSE method improves upon NUC, K-mer, and BPE for pre-training on long sequences; and 2) 3-mer tokenization leads to the highest fine-tuning performance. CSE with $k = 9$ and 3-mer tokenization are chosen as optimal encoding methods based on these results, which shall be elaborated in this section. Figure 4.8 shows the obtained F1-scores after probing and fine-tuning, both setups are explained in Section 3.6. All models are trained under the same conditions (Section 3.6). Figure 4.9 shows latent space visualizations of the validation set for all encoding methods. The findings of the comparison are explained below.

4.4.1 Three-Base Periodicity Affects Model Performance and Sequence Embeddings

Encoding methods that align with the three-base periodicity in coding RNA are shown to better distinguish coding from non-coding RNA after pre-training, due to their sensitivity to biological reading frames. These methods include K-mer Tokenization and CSE with $k \mod 3 = 0$, i.e. encoding methods that always tokenize or embed a multiple of three nucleotides per input position.

The latent space visualizations in Figure 4.9 reveal that three-base periodic encoding methods lead to more clusters in the data and a clearer distinction between pcRNA and lncRNA. The latter is also reflected by the achieved F1-scores after probing. The embedding output represents the meaning of a sequence, good models therefore assign similar embeddings to sequences with similar functions or characteristics, such as genomic elements in DNA (Dalla-Torre et al. 2023) and structural families of RNA (Penić et al. 2024; Tahmid et al. 2024). Hence, the improved distinction between pcRNA/lncRNA for encoding methods with $k \mod 3 = 0$ indicates that these configurations model the data more successfully.

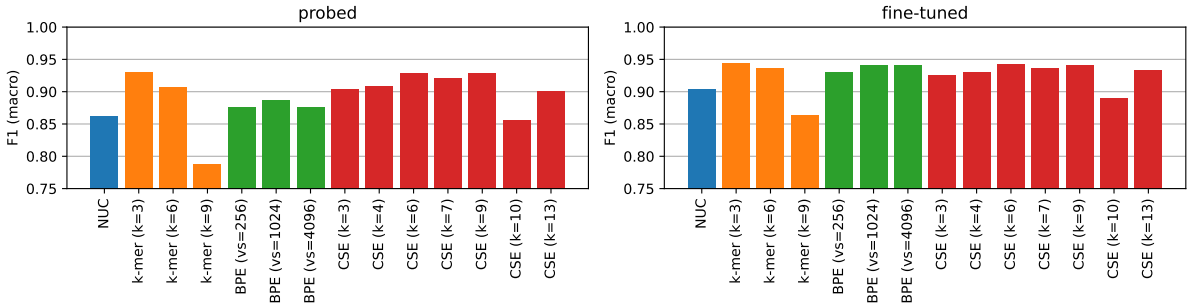

Figure 4.8: Macro-averaged F1-scores on the lncRNA classification task for different encoding methods, using probing (left) or fine-tuning (right). K-mer Tokenization with $k = 3$ leads to the highest performance for both probing (0.93) and fine-tuning (0.94) . CSE outperforms BPE when probed, meaning that CSE-based models better intrinsically distinguish pcRNA and lncRNA after pre-training. BPE models can achieve a larger performance gain than CSE when fine-tuned, leading to a better classification performance.

Figure 4.9: T-SNE visualizations of the embedding spaces of the validation set after pre-training models with different encoding methods. K-mer Tokenization with $k = 3$ and CSE with $k \in \{6, 9\}$ lead to the best distinction between pcRNA and lncRNA, reflected in the achieved F1-score when probed (0.93). Mean pooling is used to retrieve sequence-level embeddings.

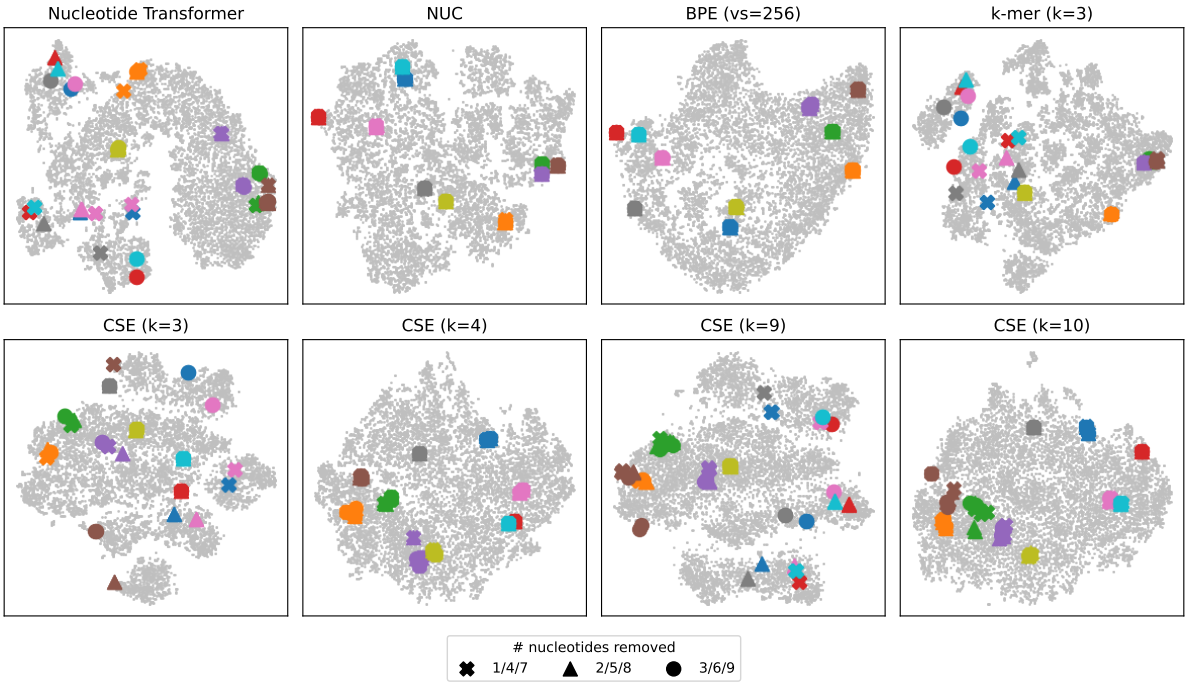

Figure 4.10: T-SNE visualization of how the embedding spaces of lncRNA-BERT models (and Nucleotide Transformer) with different encoding methods change depending on the reading frame of the input. Assessed by removing up to 9 nucleotides from 10 randomly selected sequences (indicated by color) in the validation set. The sequence-level embedding does not change for NUC, BPE, and CSE when $k \mod 3 \neq 0$. Sequence embeddings jump across three different coordinates when using K-mer Tokenization or CSE with k divisible by 3. The same is observed for the Nucleotide Transformer.

Figure 4.10 raises concerns about the biological significance of the clustering obtained by encoding methods with $k \mod 3 = 0$, as duplicate sequences jump between three groups when observed in different reading frames. Such sensitivity was anticipated for K-mer Tokenization and CSE (Section 3.3) because sequences can be observed in k distinct windows, causing k distinct embeddings at token-level. At sequence-level, good models should not generate different embeddings for the same sequence in different reading frames, since the aggregated (average) meaning of input positions has not changed. Figure 4.10 shows that this applies to models using NUC, BPE, or CSE with $k \in [4, 10]$. However, sequence-level embeddings change per reading frame when using K-mer Tokenization or CSE with k as a multiple of 3. This is shown to also apply for the Nucleotide Transformer, which uses 6-mer tokenization as encoding method.

Specifically, sequences change position in the embedding space depending on whether 1/4/7, 2/5/8 or 3/6/9 nucleotides are removed from the original sequence, alternating between 3 latent space coordinates instead of a single or k different coordinates. This is reminiscent of but not equal to the biological reading frame, which has the same periodicity but is defined by a start codon instead of the start of the sequence.

It is clear that biological reading frames in the data have an effect on the behavior of three-base periodic models. We believe these models to favor in-frame signals during training. These signals are easier to learn (and predict) because of their periodic organization (Figure 4.11) and because they do not require the model to infer k-mers from different input positions. E.g. from 'ACT TGA ACT' it is easier to learn that 'TGA' follows 'ACT' than that 'GAA' follows 'CTT' as the latter requires the model to combine multiple 3-mers. During prediction of sequences in different reading frames, the presence/absence of these easy-to-learn signals is likely to cause the observed shifts between embeddings in Figure 4.10. Note that these in-frame signals do not occur in ncRNA altogether, explaining why their embeddings are more consistent. Frameshift sensitivity is mitigated by setting $k \mod 3 \neq 0$ (e.g. $k \in [4, 7, 10]$), which breaks up the three-base periodicity. In these cases, the model is forced to combine input positions for its predictions and cannot easily develop a bias towards in-frame patterns.

We acknowledge that the sensitivity of our models towards the reading frame of the input indicates

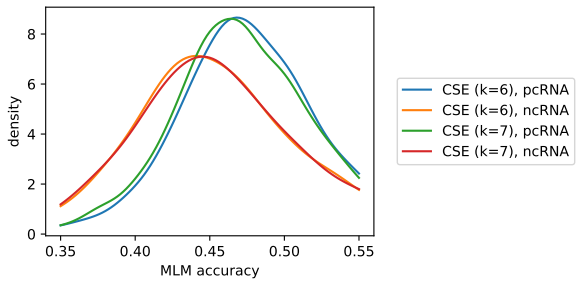

Figure 4.11: Density plot of MLM accuracy per sequence by CSE models with $k \in [6, 7]$. Both models are better at predicting pcRNA than lncRNA. Setting $k = 6$ leads to a slightly increased performance on pcRNA in comparison to $k = 7$, while the ncRNA performance is similar. The latter can be explained by the 3-base periodicity of coding RNA, which aligns better to $k = 6$ than to $k = 7$.

their understanding of the data is limited. We anticipate that a better model will recognize biologically relevant ORFs in the data and will generate consistent embeddings for every possible input reading frame. We discuss possible steps towards obtaining such a model in Chapter 5.

While the reading frame sensitivity of models with three-base periodic encoding methods might seem problematic, our results indicate that it actually helps to distinguish coding from non-coding RNA. We know from our feature-based approach that three-base periodicity can be an effective predictor of coding potential (Section 4.2.3, Figure 4.5). This explains why models with k divisible by three achieve a better probing and fine-tuning performance than models with $k \mod 3 \neq 0$ in Figure 4.8. We compare the MLM accuracy per sequence for $k = 6$ and $k = 7$ in Figure 4.11 and verify that setting $k \mod 3 = 0$ leads to a slightly better pcRNA MLM accuracy while leaving the lncRNA accuracy mostly unaffected.

4.4.2 CSE Is the Most Effective Encoding Method for Pre-Training on Long Sequences

Our results show that CSE leads to the best trainable models for encoding methods with a large sequence length reduction ($\geq 6\times$). This is concluded from Figure 4.8, where CSE and K-mer Tokenization obtain a probing F1-score of 0.93 versus 0.91 for $k = 6$, and 0.93 versus 0.79 for $k = 9$, respectively. The fine-tuning scores are also higher for these CSE and k-mer configurations.

We attribute CSE's superiority in these use cases to the complete independency between tokens in K-mer Tokenization. Learning the meaning of every token in a large vocabulary (4^k) introduces sampling efficiency issues (Sanabria et al. 2024; Z. Zhou, Ji, et al. 2023) and blows up the number of parameters (Section 3.3). In contrast, CSE sees k-mers as combinations of nucleotides and prioritizes important patterns based on the data. Figure 4.12 shows a selection of learned kernels for a pre-trained model with CSE $k = 9$, which includes both short as well as longer, more complicated patterns, the latter of which somewhat resemble biological motifs. Like the k-mer approach, BPE encodes sequences into fully independent tokens, explaining why it is outperformed by CSE when probed.

4.4.3 K-Mer and BPE Allow for Better Fine-Tuning for LncRNA Classification than CSE

CSE is outperformed on the lncRNA classification task by fine-tuned models based on 3-mer tokenization and BPE, as shown in Figure 4.8. We identify several causes supported by the results.

Figure 4.8 indicates that tokenizers work better than CSE when the token size is small and fine-tuning is allowed. CSE with $k = 3$ results in a fine-tuning F1-score of 0.926, while 3-mer and BPE (for $vs = 256$) achieve 0.944 and 0.931, respectively. Shorter tokens enable attention at a higher resolution, which may lead to improved contextualized embeddings. In addition, the vocabulary size of these methods is relatively small, diminishing the issue of sampling inefficiency and increasing the effectivity of considering tokens as independent units. During fine-tuning, these issues are of even smaller concern than during pre-training, as the model can more easily prioritize tokens that are important for the lncRNA classification task over less important ones.

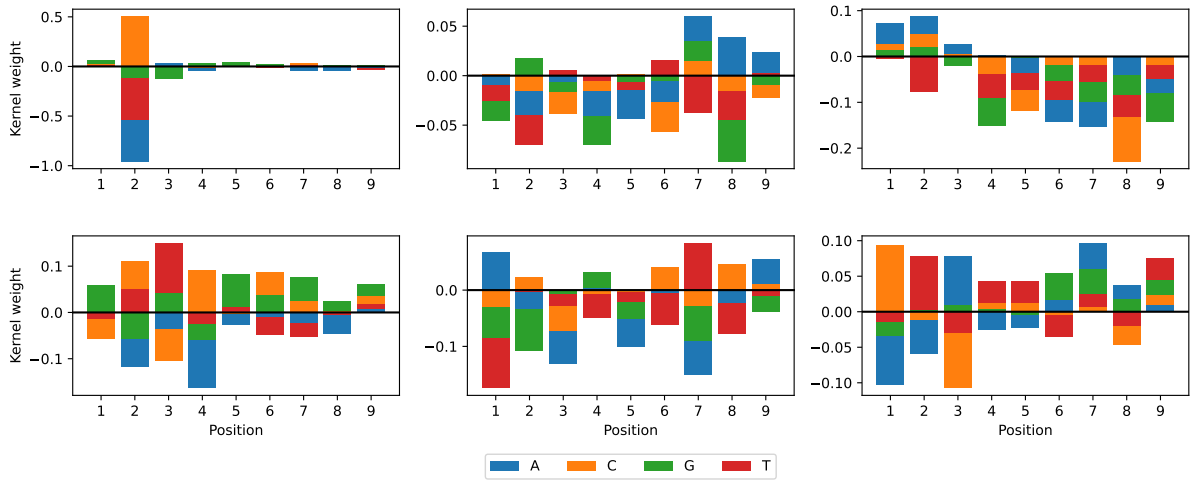

Figure 4.12: A selection of CSE ($k = 9$) kernels after pre-training. Some kernels are dedicated to recognizing nucleotides at specific positions, others match subsequences of different lengths and complexities.

Another factor that affects classification performance is sequence length coverage, which differs per encoding method as shown in Section 3.3, Figure 3.3. A longer sequence coverage allows the model to consider a larger part of the sequence in its predictions. This is a trade-off, as it can improve fine-tuning but complicate pre-training. Pre-training on longer sequences forces the model to consider a wider context in generating embeddings, possibly distracting it from important local signals. During fine-tuning, the model can more easily prioritize important input positions, which may sometimes occur near the end of a sequence. Hence, the model benefits from a longer sequence coverage. This trade-off is observed when studying the performance of CSE-based models for different k in Figure 4.8. These models benefit from larger values of k up to $k = 9$. The performance starts to deterioriate for $k > 9$.

The context length of a model with 3-mer tokenization ($3 \times 768 = 2304$) is shown to be sufficient for classifying coding potential. The biological implication here is that the coding potential of long RNA transcripts is usually inferrable from the first 2304 nucleotides. This, in combination with its small vocabulary, explains why a model with 3-mer tokenization achieves a similar fine-tuning performance as CSE with $k = 9$ (0.944 versus 0.941), despite having a smaller context length. Both configurations are demonstrated to be highly effective for both probing and fine-tuning.

4.5 Comparison to Existing NLMs

We compare lncRNA-BERT with 3-mer tokenization to previously released NLMs by visualizing their embedding spaces of the validation set in Figure 4.13. All of the models distinguish pcRNA and lncRNA to some extent, but RiNALMo, lncRNA-BERT, and DNABERT-2 seem to be more effective than BiRNA-BERT and GENA-LM. The embedding quality does not seem to be molecule-specific, as DNABERT-2 succesfully separates pcRNA from ncRNA while being trained on DNA data. BiRNA-BERT, RiNALMo, and DNABERT-2 are of equal model size ($\approx$ 110M parameters, Table 2.3) but generate different embeddings, as each of these methods has its own training procedure and sequence encoding method. The embeddings generated by lncRNA-BERT indicate its competitiveness with other NLMs on this dataset despite being the smallest model (85M parameters).

The high-quality separation of coding and non-coding RNAs in the embedding space of RiNALMo is noteworthy, as the authors claim that it has only been trained on non-coding RNA data (Penić et al. 2024). Its superiority may be attributed to the large model size (650M parameters) or the unique data sampling procedure, which involves clustering the data into groups of similar sequences, taking samples from each cluster with equal probability. Nevertheless, RiNALMo should be unfamiliar with mRNA data. The fact that it assigns distinct embeddings to mRNA indicates that the model generalizes well to unseen types of data.

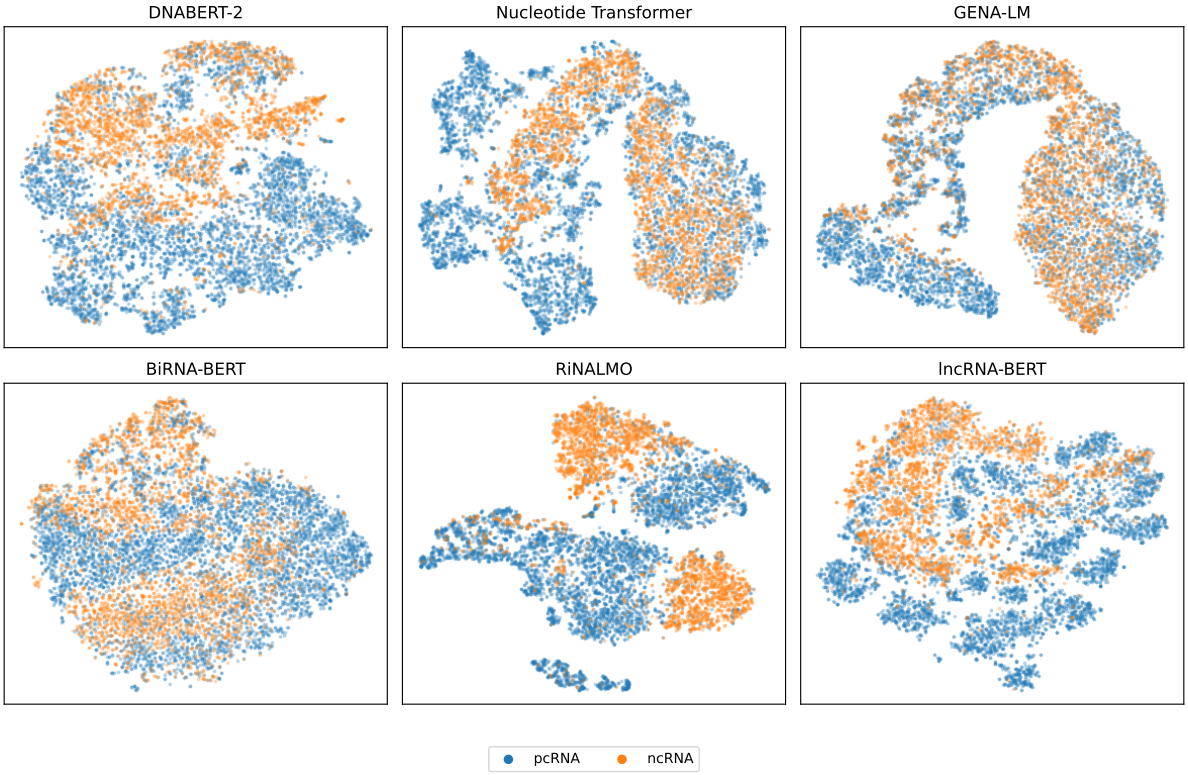

Figure 4.13: T-SNE visualizations of the embedding spaces of the validation set, generated by different NLMs. RiNALMo generates the clearest distinction between coding and non-coding transcripts, with ncRNAs clustered into two groups. LncRNA-BERT (3-mer), DNABERT-2, and BiRNA-BERT also separate the two classes. GENA-LM's embedding space seems to be less informative for lncRNA classification.

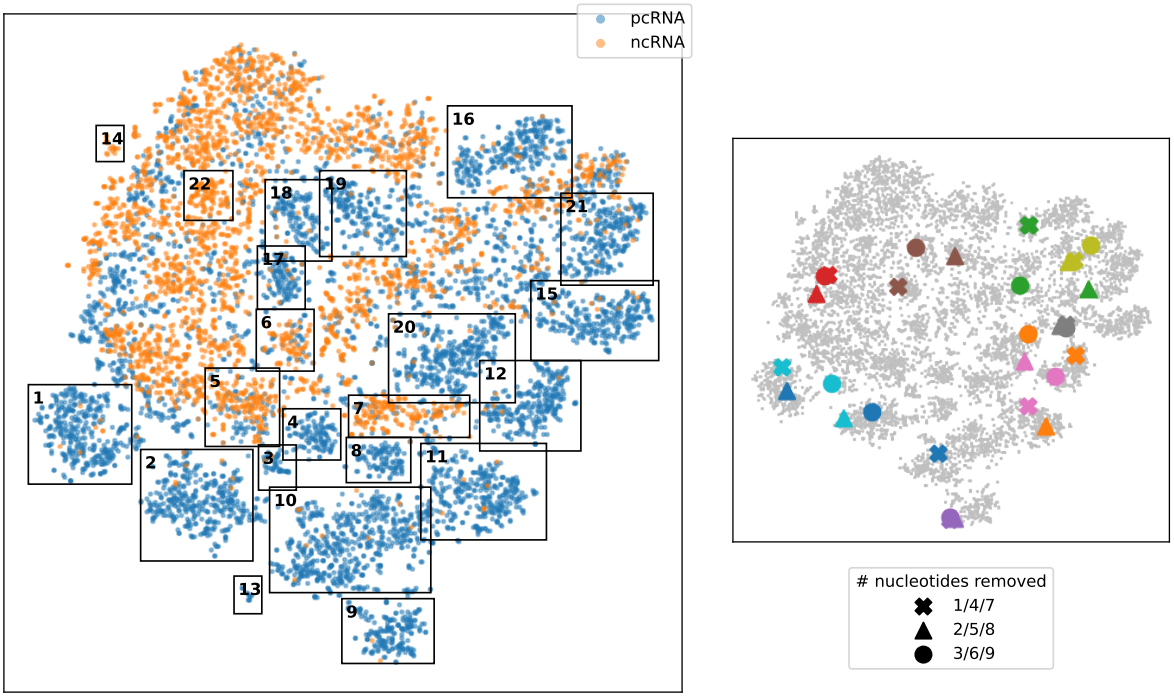

Figure 4.14: Manual labeling of clusters in the t-SNE visualization of the validation set embedding space of lncRNA-BERT with 3-mer tokenization (left) and the frameshift sensitivity of 10 randomly selected pcRNAs (right). Cluster sets $\{1, 2, 10\}$, $\{17, 18, 19\}$, and $\{11, 12, 20\}$ appear to be related, as sequences jump between different clusters within these sets when applying different reading frames to the same input. Reading frames are simulated by removing nucleotides from the start of the sequence.

4.6 Latent Space Inspection

LncRNA-BERT's latent space visualization of the validation set in Figure 4.14 (generated by the 3-mer tokenization model) shows distinct clusters of pcRNAs, which are mostly well-separated from lncRNA. LncRNA clusters also exist, although these are less well distinguished (e.g. cluster 5, 6, 7, 14, and 22). Observing more clusters in pcRNA aligns with Section 3.3 and indicates a preference of lncRNA-BERT towards pcRNA. As discussed in that same section and shown in Figure 4.14, most pcRNA sequences iterate between three different groups depending on their input reading frame.

Aside from that, it is unclear what exactly the clusters represent. The model assigns similar contextualized embeddings to specific sets of sequences, but its black box nature does not facilitate a direct interpretation of the embedding space. We know from Figure 4.7 that our RNAcentral model recognizes different types of RNA, which motivates the search for relevance in the latent space of our human RNA model. Unlike the ArchiveII data visualized in Figure 4.7, the RNA data in our validation set is not classified into different types. Instead, we use Gene Ontology (GO) annotations to investigate the embedding space.

We first confirm that the clustering is not directly based on sequence length, although Figure 4.15 points out that a length gradient exists. PcRNA and lncRNA sequences at the upper half of the visualization are shorter in comparison to the lower half.

Figure 4.15 also indicates a significantly higher MLM accuracy for sequences in cluster 9. We do not know what is causing the model to perform particularly well on these sequences. Figure 4.14 shows that a sequence in this cluster does not change its position when frame-shifted. This hints toward the resolution of the frameshift sensitivity problem when MLM accuracy is improved.

The contextualized embedding space does not strictly correspond to sequence similarity, as embeddings are based on the meaning of a sequence instead of its composition (although the two are related). The mean pairwise Euclidian distance between 6-mer spectra of sequences within each cluster is used as a proxy for intra-cluster sequence similarity. The average pairwise 6-mer distance between all sequences in the dataset equals 4.57e-02. The mean distance for each cluster is listed in Table 4.2, showing that most clusters have a lower mean distance than the dataset average. Therefore, we conclude that these

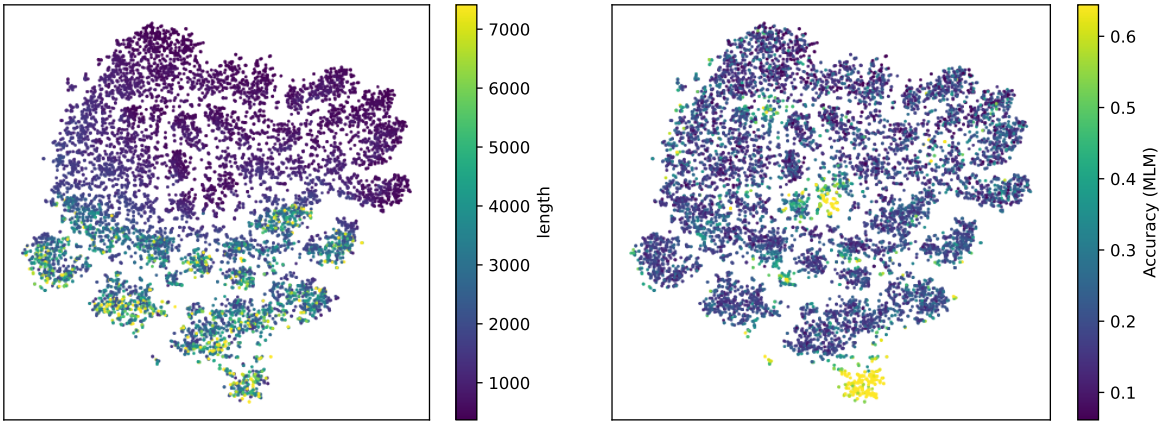

Figure 4.15: T-SNE visualization of the sequence-level embedding space of the validation set by lncRNA-BERT w.r.t. sequence length (left) and MLM accuracy (right). Clusters in the bottom half of the embedding space contain sequences of varying lengths, but clusters in the upper half are exclusively comprised of short sequences. The model obtains low MLM accuracies (<0.3), except for two clusters with a significantly higher score (≈0.6).

sequences are more similar to each other than to the rest of the data. Some clusters (6, 15, 17, 19, 21, 22) are exceptions to this conclusion and exhibit a relatively high intra-cluster 6-mer distance. This emphasizes that the embedding space is not merely a representation of sequence similarity. Note that using pairwise alignment distances to express sequence similarity would be preferred over using 6-mer spectra, but this would cost significant computational resources to calculate.

We use the online g:GOSt tool from g:Profiler (`https://biit.cs.ut.ee/gprofiler/gost`) to perform a functional enrichment analysis on the lists of gene names for each cluster labeled in Figure 4.14. The most significantly enriched GO terms are listed in Table 4.2. Only driver terms are reported to prioritize the GOs that induce other significant terms, these are identified using a greedy filtering algorithm built into g:GOSt. We assess the validity of this approach by running g:GOSt five times with 200 randomly selected gene names from the validation set. This results in statistically significant findings for 5/5 repetitions, although never resulting in $P_{adj} < 10^{-6}$. Hence, we shall focus only on highly significant findings.

Most of the enriched GO terms are broad (e.g. 'protein binding', 'cytoplasm') or of low limited statistical significance The most significant findings are those of clusters 3, 10, and 11, respectively identifying 'regulation of DNA-templated transcription', 'anatomical structure development', and 'regulation of developmental process' as enriched biological processes, with a P_{adj} of 6.75e-37, 2.96e-14, and 2.24e-10. These findings support the biological relevance of different clusters in the latent space. Cluster 9, which has a high MLM accuracy (Figure 4.15), seems to contain pcRNAs that are related to 'nervous system development'. Table 4.2 indicates that clusters with highly significant enriched GO terms often exhibit a low mean 6-mer distance. Sequence similarity may thus have affected the enrichment results. On the other hand, some clusters with highly diverse sequences (16, 19) also result in significant findings, whereas some clusters with highly similar sequences (7, 8) do not.

The enrichment results can also be used to further investigate the frameshift sensitivity problem of three-base periodic encoding methods, as explained in Section 4.4.1. Figure 4.14 identifies the following clusters sets where sequences seem to jump within depending on their input reading frame: $\{1, 2, 10\}$, $\{17, 18, 19\}$, and $\{11, 12, 20\}$. Similar GO terms are enriched for clusters 1 and 2 ('organelle organization'), which may indicate that these clusters are frameshifted variants of each other. Contrastingly, cluster sets $\{17, 18, 19\}$ and $\{11, 12, 20\}$ report different GO terms. Different reading frames may thus cause different types of signals to be extracted.

Clusters containing lncRNA transcripts do not result in highly significantly enriched GO terms, as lncRNAs are less well annotated than pcRNAs. However, our pcRNA-related findings indicate that these clusters may also be of biological relevance.

ID	Sequence distance	Molecular Function	P_{adj}	Biological Process	P_{adj}	Cellular Component	P_{adj}
1	2.71e-02	protein binding	1.68e-15	organelle organization	3.47e-07	cytoplasm	1.21e-14
2	2.66e-02	protein binding	1.15e-06	organelle organization	4.47e-08	cytoplasm	5.47e-16
3	3.25e-02	DNA-binding transcription factor activity	6.06e-53	regulation of DNA-templated transcription	6.75e-37	nucleus	1.86e-15
4	2.68e-02	protein binding	2.30e-03	regulation of biological process	2.31e-03	cytoplasm	1.92e-04
5	3.03e-02					TIM22 mitochondrial import inner membrane insertion complex	4.99e-02
6	4.68e-02	DNA-binding transcription factor activity, RNA polymerase II-specific	3.60e-02				
7	3.19e-02						
8	2.74e-02						
9	2.69e-02	protein binding	1.70e-05	nervous system development	1.51e-07	cytoplasm	1.87e-06
10	2.78e-02	protein binding	1.26e-13	response to stimulus	2.96e-14	cytoplasm	4.58e-21
11	3.05e-02	protein binding	5.23e-06	anatomical structure development	2.24e-10	cytoplasm	3.61e-05
12	3.14e-02	protein binding	2.15e-05	regulation of developmental process	1.23e-08	cell periphery	2.57e-07
13	2.75e-02			regulation of RNA splicing	2.03e-04		
14	4.43e-02			central nervous system interneuron axonogenesis	4.95e-02		
15	5.58e-02	protein binding	4.23e-05			cytoplasm	1.43e-09
16	6.01e-02	protein binding	3.56e-06	organonitrogen compound metabolic process	2.70e-05	cytoplasm	2.98e-18
17	5.45e-02			ncRNA processing	5.70e-03	cytoplasm	2.77e-06
18	5.63e-02			protein metabolic process	6.67e-03	cytoplasm	8.15e-06
19	5.79e-02	protein binding	9.13e-06	transport along microtubule	6.71e-04	cytoplasm	3.17e-10
20	3.21e-02	protein binding	9.30e-05	response to abiotic stimulus	6.43e-04	cytoplasm	6.46e-07
21	5.88e-02	protein binding	4.05e-05	regulation of cellular component organization	8.33e-03	cytoplasm	2.20e-11
22	5.58e-02						

Table 4.2: Most significantly enriched gene ontology terms for clusters in the validation set as visualized and labeled in Figure 4.14. Statistical enrichment analysis is carried out by inputting the transcript-associated gene names to g:Profiler g:GOSt with default settings. Reporting only driver terms with a significant enrichment. Sequence similarity is expressed as the mean pairwise Euclidian distance between 6-mer spectra of sequences within a cluster.

Chapter 5

Discussion

The human RNA language model proposed in this thesis, lncRNA-BERT, is demonstrated to obtain state-of-the-art performance in distinguishing coding from long non-coding RNA (Section 4.1). The fine-tuned lncRNA-BERT model outperforms five of the six previously published lncRNA classifiers in our comparison, with LncADeep being the only previous method that obtains a higher F1-score than both lncRNA-BERT configurations on one of the three test sets. In addition, lncNRA-BERT generates an improved distinction between pcRNA and lncRNA in its embedding space in comparison to NLMs of equal model size (Section 4.5). *This answers Research Question 1.*

A large set of RNA predictory features, originating from various previous studies, has been re-implemented in the lncRNA-Py package. We show that some of these features (k-mer frequencies, entropy, three-base periodicity) indicate the presence of unique linguistic patterns for mRNA/lncRNA, strongly motivating the applicability of an RNA language model for this task. We also establish two feature-based machine learning baselines to compete with lncRNA-BERT (Section 4.2). LncRNA-RF, our random forest baseline model with relative Recursive Feature Selection, achieves a higher F1-score than all of the classifiers from previous works, and also outperforms lncRNA-BERT on all three test sets. This indicates that despite the potential of sequence-based models, lncRNA classifiers benefit from incorporating sequence-extrinsic information. *This answers Research Question 2.*

In comparison to other NLMs, lncRNA-BERT is specialized in human mRNA and lncRNA, whereas alternative RNA foundation models are often pre-trained on the cross-species RNAcentral dataset, which contains mostly short ncRNAs. We show that pre-training on human RNA from GENCODE, RefSeq, and NONCODE results in learning a sequence-intrinsic distinction between pcRNA and lncRNA, leading to a higher downstream performance in comparison to using RNAcentral (Section 4.3). The learned embeddings are indicated to contain biologically relevant information beyond coding potential, indicating lncRNA-BERT's potential for fine-tuning on different tasks (Section 4.6). *This answers Research Question 3.*

Furthermore, we introduce Convolutional Sequence Encoding (CSE), which encodes long nucleotide sequences in a more effective and parameter-efficient way than K-mer Tokenization or BPE (Section 4.4). CSE obtains its increased effectivity for long sequences by convolving position weight matrices to directly embed k-mers into a high-dimensional representation, instead of considering them as fully independent tokens. In an extensive comparison, 3-mer tokenization yields comparable lncRNA classification results, indicating that full sequence length coverage is not a requirement for this task. Nevertheless, our lncRNA-BERT model with CSE and $k = 9$ accommodates sequences of almost 7000 nt while using the standard BERT architecture as base model. An additional result of our encoding method comparison is the discovery of a frameshift sensitivity issue when using three-base periodic encoding methods, which also applies to the widely-used Nucleotide Transformer (Section 4.4.1). *This answers Research Question 4.*

A reflection upon the results and applied methods is provided below. Section 5.1 addresses the benefits and limitations of the NLM approach for discriminating between mRNA/lncRNA. We then discuss how data clustering can improve our methodology in Section 5.2. Section 5.3 comments on the competitiveness of our model in comparison to other available NLMs. Finally, we provide recommendations for future work on NLMs in Section 5.4.

5.1 Benefits and Limitations of NLMs for LncRNA Classification

The NLM approach to distinguishing coding from non-coding RNA is shown to be effective, but fails to consistently outperform the feature-based lncRNA-RF and LncADeep algorithms. This questions the added value of using NLMs for this task. In machine learning, the principle of Occam's razor states that when two models achieve a similar performance, the simpler model is preferred. This preference towards simpler is mainly motivated by the notion that models with less parameters are less likely to overfit on the data. LncRNA-RF has 14× fewer trainable parameters (6M vs 89M) than lncRNA-BERT, but achieves a similar F1-score (0.95, on GENCODE). Based on the obtained performances shown in Figure 4.1, we therefore cannot conclude that using the sequence-based lncRNA-BERT is beneficial over using the feature-based lncRNA-RF.

A possible explanation for why our NLM does not always outperform existing lncRNA classification algorithms is that some of these methods might have already reached the best possible performance. Judging from the F1-scores (0.94, on CPAT) obtained by the highly simplistic lncRNA-LR algorithm (11 parameters), we believe that achieving a seemingly reasonable performance on this task is trivial. At the same time, we anticipate a performance plateau due to the lack of a strict binary separation between pcRNA and lncRNA. For example, lncRNA contains short ORFs (Figure 4.2) which may encode functional micro-peptides (Pang et al. 2018) and mRNA can have ncRNA-like regulatory functions (Kloc et al. 2011; J. Li et al. 2020; Mustoe et al. 2018). The true function of a transcript may very well be context-dependent, e.g. varying between tissues or developmental stages. This information is not incorporated in any lncRNA classifier, as such data is not available. The annotation systems in databases like RefSeq and GENCODE falsely assume an unambiguous distinction between the two classes, complicating the training and testing of ML models. Consequently, it is impossible to reach 100% accuracy without overfitting to the human labeling system. Given the vast number of lncRNA classifiers that have been published in the past 15 years (Table 2.2), it does not seem unlikely that their maximal performance (F1≈0.95) corresponds to the described plateau, which would explain why we do not outperform them.

A benefit of lncRNA-BERT over traditional classifiers is that its pre-training procedure does not utilize human-assigned labels but still clearly distinguishes coding from non-coding RNA, based solely on the transcript sequence. The clusterings in Figure 4.14 show that lncRNA-BERT largely succeeds in separating pcRNA and lncRNA in the dataset. Some pcRNA clusters are polluted with lncRNAs, and vice versa. These data points could indicate inaccuracies of our model, but also motivate reconsideration of the database-assigned label. Using an unsupervised approach might lead to a more nuanced view on the two RNA classes. This idea could be pursued in future work by using an NLM like lncRNA-BERT to assign novel RNA classes, e.g. with the TURTLE framework (Gadetsky et al. 2024).

5.2 Data Clustering Holds Potential For Future Work

Many lncRNA classification studies, including ours, suffer from overlap between train and test sets (D. Singh et al. 2022). This introduces bias in the comparison between different classification algorithms. Our method of mitigating this bias, clustering with CD-HIT, may also positively affect pre-training in future work.

Each previous lncRNA classifier has a unique definition of train and test datasets, which means that our test sequences might have been part of their training data (e.g. PredLncGF-Stack was trained on GENCODE). To guarantee a fair evaluation, one should re-train each of the compared methods with our training set. This is a time-consuming effort that would have led to the inclusion of fewer algorithms, also because not all methods offer a re-training option in their official software release (e.g. PredLnc-GFStack, LncADeep). This type of overlap positively affects the performance of previous lncRNA classifiers relative to ours. We decide to neglect this bias, as it does not favor our own method.

Data redundancy removal is required to mitigate a second type of train/test overlap, caused by duplicate or similar sequences present in both the training and evaluation datasets. Similar to our approach, previous works have used clustering algorithms like CD-HIT and MMSeqs2 to remove sequences that occur multiple times in the same dataset (Feng et al. 2023; S. Liu et al. 2019), preventing them from being present in both train and test data.

The application of a clustering algorithm may also be the key to properly including cross-species data during pre-training, due to an increased data diversity. Section 4.3 shows that pre-training on RNAcentral data does not lead to a well-performing model on mRNA/lncRNA, exposing a potential

weakness of some existing RNA NLMs. Nevertheless, models pre-trained on RNAcentral achieve high MLM accuracies on specific overrepresented types of RNA (Figure 4.7). Previous works have shown that NLMs benefit from multi-species data (Dalla-Torre et al. 2023; Z. Zhou, Ji, et al. 2023). Moreover, the RNAcentral dataset contains significantly more sequences than our pre-training set (37M vs 0.5M). We therefore anticipate that removing the redundancy from RNAcentral with a clustering algorithm might mitigate the currently faced issues and lead to a superior RNA foundation model. To ensure optimal performance on the lncRNA classification task, one would also have to include pcRNA data during pre-training. The pre-training procedure could sample from clusters obtained by CD-HIT/MMSeqs2 from the combined dataset, similar to the approach taken by RiNALMo (Penić et al. 2024).

5.3 Addressing the Competitiveness of LncRNA-BERT with Other NLMs

Section 4.5 shows that, based on the obtained embeddings, lncRNA-BERT is better adapted to human mRNA/lncRNA data than most other NLMs included in our analysis. This is not a surprising observation as lncRNA-BERT was specifically trained on this type of data, and its encoding method was optimized for lncRNA classification (Section 4.4). The results highlight how the behavior of NLMs will change depending on what type of DNA/RNA data is given to them during training and inference (also seen in Section 4.3). For a more thorough comparison, we recommend fine-tuning other NLMs for the lncRNA classification task, as well as fine-tuning all methods (including lncRNA-BERT) for alternative tasks that focus on long RNA sequences, such as splice site detection.

Other NLMs utilize several architectural advancements that are currently not implemented by lncRNA-BERT but could lead to improvements. In this work, we specifically focus on the choice of encoding method to increase the accepted sequence length, while keeping the BERT architecture as is. Figures 4.8 and 4.9 show that CSE can adapt to longer sequence lengths more effectively than K-mer Tokenization and BPE. Section 4.5 indicates that lncRNA-BERT obtains a pcRNA/lncRNA embedding split comparable to that of RiNALMo despite being 8× smaller in size. Future improvements to lncRNA-BERT can be realized by incorporating the advancements proposed in other NLMs. These include Rotary Positional Embeddings (RoPE, Dalla-Torre et al. 2023; Penić et al. 2024; Su et al. 2021; X. Wang et al. 2023) or Attention with Linear Biases (ALiBi, Press et al. 2021; Tahmid et al. 2024; Z. Zhou, Ji, et al. 2023) for improved generalization to longer sequences, Flash Attention for an increased efficiency (Dao et al. 2022; Penić et al. 2024; Z. Zhou, Ji, et al. 2023), and SwiGLU activations for better training convergence (Dalla-Torre et al. 2023; Penić et al. 2024). Making these modifications and/or increasing the size of the BERT architecture by modifying its hyperparameters is likely to further increase performance.

5.4 Recommendations for Improving NLMs

The findings in this thesis indicate that lncRNA-BERT and other existing NLMs have a limited understanding of the data. We anticipate that NLMs can be significantly improved in future work by continuing to study effective encoding methods and training on larger and more diverse datasets. Previous studies have shown that NLMs can obtain state-of-the-art performance in tasks such as splice site detection (Dalla-Torre et al. 2023) and chromatin profile prediction (Fishman et al. 2023). In our work, we show that lncRNA-BERT achieves state-of-the-art performance in lncRNA classification and generate biologically informative embeddings. Nevertheless, the low MLM accuracy (< 0.5, Figure 4.11) indicates that the model makes a substantial amount of mistakes in predicting masked nucleotides (or tokens). Furthermore, the embedding space of lncRNA-BERT suffers from a reading frame bias (Figure 4.10) and other NLMs do not convincingly distinguish additional lncRNA clusters beyond the mRNA/lncRNA split, despite literature indicating that multiple types of lncRNA exist (Figure 1.1).

One way to improve NLMs would be to further pursue the search for an optimal sequence encoding method, i.e. finding the best definition of a linguistic unit in DNA/RNA. While developments such as ALiBi and Flash Attention can extend the context length of a transformer model with Nucleotide-level Tokenization, we advise a more compressive encoding method for a more efficient NLM. Section 4.4 identifies 3-mer tokenization as optimal encoding technique for lncRNA classification, but a comparison on different downstream tasks is required to come to a general conclusion. Because lncRNA classification is relatively simple (Section 5.1), it may not be the most suitable task to highlight differences between the compared techniques. In addition, it may induce bias towards three-base periodic encoding meth-

ods (Section 4.4.1). Finally, alternative downstream tasks might require the processing of longer input sequences, which favor CSE. The non-existence of a single optimal method for all tasks is referred to as the 'no free lunch' theorem in machine learning.

Our CSE method is shown to be more effective and parameter efficient than BPE and K-mer Tokenization for pre-training on long sequences, but may need to be improved for utilization in future NLMs. Despite numerous efforts, we could not find a way to achieve the same fine-tuning performance gain for CSE (relative to its probing performance) as for BPE and K-mer tokenization. It is unclear which factors attribute to the observed instability in CSE's fine-tuning, although the highly flexible kernels in the input layer may play a role. Solving the fine-tuning stability issue of CSE may also increase its pre-training performance. Another shortcoming of CSE is that it does not outperform K-mer tokenization for $k = 3$, indicating that tokens are more suitable than our convolutional encodings when k is low. CSE and K-mer Tokenization encode the same number of nucleotides per input position and should therefore, in theory, be able to achieve the same performance. This may be achieved through architectural optimizations of the CSE layer.

We anticipate that NLMs, especially RNA LMs, can be greatly improved by increasing the amount of pre-training data and the genetic diversity within it. Adding intra- and inter-species diversity to the pre-training task can cause a model to generalize over subtle signals between different individuals and phylogenetic signals between species. These beneficial effects have been demonstrated for other NLMs, e.g. the Nucleotide Transformer is able to perform variant prioritization (Dalla-Torre et al. 2023) and GENA-LM can be used for taxonomic classification (Fishman et al. 2023). It is unlikely that NLMs will be trained with genomic data from a large ($>> 1,000$) number of individuals within the near future, as this data is not (publicly) available and using it for this purpose would bring up several ethical and privacy-related concerns. Nevertheless, existing NLMs can be improved by incorporating more of the available data. Specifically for RNA, the RNAcentral dataset is limited to non-coding RNA, even though resources like Ensembl also contain multi-species mRNA data. Using this data in combination with a proper clustering algorithm will likely lead to more informative embeddings than currently generated by RiNALMo and lncRNA-BERT.

Perhaps a ChatGPT-like breakthrough, which has not occurred for NLMs so far, can only be achieved with an advanced sequence encoding method and a significant increase in data size and diversity. Such a breakthrough could then lead to novel discoveries in genetics, such as an improved characterization of long non-coding RNA.

Bibliography

Achawanantakun, Rujira et al. (Aug. 2015). "LncRNA-ID: Long non-coding RNA IDentification using balanced random forests". In: *Bioinformatics* 31.24, pp. 3897–3905. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv480.

Akiyama, Manato and Yasubumi Sakakibara (Jan. 2022). "Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning". In: *NAR Genomics and Bioinformatics* 4.1. ISSN: 2631-9268. DOI: 10.1093/nargab/lqac012.

Arrial, Roberto T, Roberto C Togawa, and Marcelo de M Brigido (Aug. 2009). "Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus Paracoccidioides brasiliensis". In: *BMC Bioinformatics* 10.1. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-239.

Avsec, Žiga et al. (Oct. 2021). "Effective gene expression prediction from sequence by integrating long-range interactions". In: *Nature Methods* 18.10, pp. 1196–1203. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01252-x.

Baek, Junghwan et al. (May 2018). "LncRNAnet: long non-coding RNA identification using deep learning". In: *Bioinformatics* 34.22. Ed. by Alfonso Valencia, pp. 3889–3897. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bty418.

Barriocanal, Marina et al. (Jan. 2015). "Long Non-Coding RNA BST2/BISPR is Induced by IFN and Regulates the Expression of the Antiviral Factor Tetherin". In: *Frontiers in Immunology* 5. ISSN: 1664-3224. DOI: 10.3389/fimmu.2014.00655.

Bosco, Giosué Lo and Mattia Antonino Di Gangi (2017). "Deep Learning Architectures for DNA Sequence Classification". In: *Fuzzy Logic and Soft Computing Applications*. Springer International Publishing, pp. 162–171. DOI: 10.1007/978-3-319-52962-2_14.

Busia, Akosua et al. (June 2018). "A deep learning approach to pattern recognition for short DNA sequences". In: DOI: 10.1101/353474.

Camargo, Antonio P et al. (Jan. 2020). "RNAsamba: neural network-based assessment of the protein-coding potential of RNA sequences". In: *NAR Genomics and Bioinformatics* 2.1. ISSN: 2631-9268. DOI: 10.1093/nargab/lqz024.

Cao, Lei et al. (Aug. 2020). "PreLnc: An Accurate Tool for Predicting lncRNAs Based on Multiple Features". In: *Genes* 11.9, p. 981. ISSN: 2073-4425. DOI: 10.3390/genes11090981.

Cesana, Marcella et al. (Oct. 2011). "A Long Noncoding RNA Controls Muscle Differentiation by Functioning as a Competing Endogenous RNA". In: *Cell* 147.2, pp. 358–369. ISSN: 0092-8674. DOI: 10.1016/j.cell.2011.09.028.

Chen, Jiayang et al. (2022). *Interpretable RNA Foundation Model from Unannotated Data for Highly Accurate RNA Structure and Function Predictions*. DOI: 10.48550/ARXIV.2204.00300.

Crick, F H (1958). "On protein synthesis". en. In: *Symp. Soc. Exp. Biol.* 12, pp. 138–163.

Dalla-Torre, Hugo et al. (Jan. 2023). *The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics*. DOI: 10.1101/2023.01.11.523679.

Dao, Tri et al. (2022). *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. DOI: 10.48550/ARXIV.2205.14135.

Devlin, Jacob et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/ARXIV.1810.04805.

Dosovitskiy, Alexey et al. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. DOI: 10.48550/ARXIV.2010.11929.

Ender, Christine and Gunter Meister (June 2010). "Argonaute proteins at a glance". In: *Journal of Cell Science* 123.11, pp. 1819–1823. ISSN: 0021-9533. DOI: 10.1242/jcs.055210.

Fan, Chuannan et al. (June 2023). "The lncRNA LETS1 promotes TGF-–induced EMT and cancer cell migration by transcriptionally activating a TR1-stabilizing mechanism". In: *Science Signaling* 16.790. ISSN: 1937-9145. DOI: 10.1126/scisignal.adf1947.

Fan, Xiao-Nan and Shao-Wu Zhang (2015). "lncRNA-MFDL: identification of human long non-coding RNAs by fusing multiple features and using deep learning". In: *Molecular BioSystems* 11.3, pp. 892–897. ISSN: 1742-2051. DOI: 10.1039/c4mb00650j.

Fan, Xiao-Nan, Shao-Wu Zhang, et al. (July 2020). "lncRNA_Mdeep: An Alignment-Free Predictor for Distinguishing Long Non-Coding RNAs from Protein-Coding Transcripts by Multimodal Deep Learning". In: *International Journal of Molecular Sciences* 21.15, p. 5222. ISSN: 1422-0067. DOI: 10.3390/ijms21155222.

Feng, Hongqi et al. (2023). "LncCat: An ORF attention model to identify LncRNA based on ensemble learning strategy and fused sequence information". In: *Computational and Structural Biotechnology Journal* 21, pp. 1433–1447. ISSN: 2001-0370. DOI: 10.1016/j.csbj.2023.02.012.

Fickett, James W. (1982). "Recognition of protein coding regions in DNA sequences". In: *Nucleic Acids Research* 10.17, pp. 5303–5318. ISSN: 1362-4962. DOI: 10.1093/nar/10.17.5303.

Fishman, Veniamin et al. (June 2023). *GENA-LM: A Family of Open-Source Foundational DNA Language Models for Long Sequences*. DOI: 10.1101/2023.06.12.544594.

Frankish, Adam et al. (Nov. 2022). "GENCODE: reference annotation for the human and mouse genomes in 2023". In: *Nucleic Acids Research* 51.D1, pp. D942–D949. ISSN: 1362-4962. DOI: 10.1093/nar/gkac1071.

Fu, Limin et al. (Oct. 2012). "CD-HIT: accelerated for clustering the next-generation sequencing data". In: *Bioinformatics* 28.23, pp. 3150–3152. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bts565.

Gadetsky, Artyom, Yulun Jiang, and Maria Brbic (2024). *Let Go of Your Labels with Unsupervised Transfer*. DOI: 10.48550/ARXIV.2406.07236.

Guo, Jin-Cheng et al. (May 2019). "CNIT: a fast and accurate web tool for identifying protein-coding and long non-coding transcripts based on intrinsic sequence composition". In: *Nucleic Acids Research* 47.W1, W516–W522. ISSN: 1362-4962. DOI: 10.1093/nar/gkz400.

Han, Siyu, Yanchun Liang, Ying Li, et al. (2016a). "Lncident: A Tool for Rapid Identification of Long Noncoding RNAs Utilizing Sequence Intrinsic Composition and Open Reading Frame Information". In: *International Journal of Genomics* 2016, pp. 1–11. ISSN: 2314-4378. DOI: 10.1155/2016/9185496.

— (2016b). "Long Noncoding RNA Identification: Comparing Machine Learning Based Tools for Long Noncoding Transcripts Discrimination". In: *BioMed Research International* 2016, pp. 1–14. ISSN: 2314-6141. DOI: 10.1155/2016/8496165.

Han, Siyu, Yanchun Liang, Qin Ma, et al. (July 2018). "LncFinder: an integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property". In: *Briefings in Bioinformatics* 20.6, pp. 2009–2027. ISSN: 1477-4054. DOI: 10.1093/bib/bby065.

He, Shujun et al. (Nov. 2023). "Nucleic Transformer: Classifying DNA Sequences with Self-Attention and Convolutions". In: *ACS Synthetic Biology* 12.11, pp. 3205–3214. ISSN: 2161-5063. DOI: 10.1021/acssynbio.3c00154.

Helaly, Marwah A., Sherine Rady, and Mostafa M. Aref (Oct. 2019). "Convolutional Neural Networks for Biological Sequence Taxonomic Classification: A Comparative Study". In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, pp. 523–533. DOI: 10.1007/978-3-030-31129-2_48.

Hill, Steven T et al. (July 2018). "A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential". In: *Nucleic Acids Research* 46.16, pp. 8105–8113. ISSN: 1362-4962. DOI: 10.1093/nar/gky567.

Hu, Long et al. (Sept. 2016). "COME: a robust coding potential calculation tool for lncRNA identification and characterization based on multiple features". In: *Nucleic Acids Research* 45.1, e2–e2. ISSN: 1362-4962. DOI: 10.1093/nar/gkw798.

Ito, Eric Augusto et al. (June 2018). "BASiNET—BiologicAl Sequences NETwork: a case study on coding and non-coding RNAs identification". In: *Nucleic Acids Research* 46.16, e96–e96. ISSN: 1362-4962. DOI: 10.1093/nar/gky462.

Ji, Yanrong et al. (Feb. 2021). "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome". In: *Bioinformatics* 37.15. Ed. by Janet Kelso, pp. 2112–2120. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btab083.

Jumper, John et al. (July 2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.

Kang, Yu-Jian et al. (May 2017). "CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features". In: *Nucleic Acids Research* 45.W1, W12–W16. ISSN: 1362-4962. DOI: 10.1093/nar/gkx428.

Kloc, Malgorzata, Victor Foreman, and Sriyutha A. Reddy (Nov. 2011). "Binary function of mRNA". In: *Biochimie* 93.11, pp. 1955–1961. ISSN: 0300-9084. DOI: 10.1016/j.biochi.2011.07.008.

Kong, Lei et al. (July 2007). "CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine". In: *Nucleic Acids Research* 35.suppl$_2$, W345–W349. ISSN: 0305-1048. DOI: 10.1093/nar/gkm391.

Li, Aimin, Junying Zhang, and Zhongyin Zhou (Sept. 2014). "PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme". In: *BMC Bioinformatics* 15.1. ISSN: 1471-2105. DOI: 10.1186/1471-2105-15-311.

Li, Jing, Xuan Zhang, and Changning Liu (2020). "The computational approaches of lncRNA identification based on coding potential: Status quo and challenges". In: *Computational and Structural Biotechnology Journal* 18, pp. 3666–3677. ISSN: 2001-0370. DOI: 10.1016/j.csbj.2020.11.030.

Li, Minghua and Chun Liang (Nov. 2022). "LncDC: a machine learning-based tool for long non-coding RNA detection from RNA-Seq data". In: *Scientific Reports* 12.1. ISSN: 2045-2322. DOI: 10.1038/s41598-022-22082-7.

Li, Siting et al. (July 2021). "Long noncoding RNA HOTAIR interacts with Y-Box Protein-1 (YBX1) to regulate cell proliferation". In: *Life Science Alliance* 4.9, e202101139. ISSN: 2575-1077. DOI: 10.26508/lsa.202101139.

Lin, Michael F., Irwin Jungreis, and Manolis Kellis (June 2011). "PhyloCSF: a comparative genomics method to distinguish protein coding and non-coding regions". In: *Bioinformatics* 27.13, pp. i275–i282. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr209.

Liu, Jinfeng, Julian Gough, and Burkhard Rost (Apr. 2006). "Distinguishing Protein-Coding from Non-Coding RNAs through Support Vector Machines". In: *PLoS Genetics* 2.4. Ed. by Judith Blake et al., e29. ISSN: 1553-7404. DOI: 10.1371/journal.pgen.0020029.

Liu, Shuai et al. (Sept. 2019). "PredLnc-GFStack: A Global Sequence Feature Based on a Stacked Ensemble Learning Method for Predicting lncRNAs from Transcripts". In: *Genes* 10.9, p. 672. ISSN: 2073-4425. DOI: 10.3390/genes10090672.

Mattick, John S. et al. (Jan. 2023). "Long non-coding RNAs: definitions, functions, challenges and recommendations". In: *Nature Reviews Molecular Cell Biology* 24.6, pp. 430–447. ISSN: 1471-0080. DOI: 10.1038/s41580-022-00566-8.

Meng, Jun, Zheng Chang, et al. (2019). "lncRNA-LSTM: Prediction of Plant Long Non-coding RNAs Using Long Short-Term Memory Based on p-nts Encoding". In: *Intelligent Computing Methodologies*. Springer International Publishing, pp. 347–357. ISBN: 9783030267667. DOI: 10.1007/978-3-030-26766-7_32.

Meng, Jun, Qiang Kang, et al. (May 2021). "PlncRNA-HDeep: plant long noncoding RNA prediction using hybrid deep learning based on two encoding styles". In: *BMC Bioinformatics* 22.S3. ISSN: 1471-2105. DOI: 10.1186/s12859-020-03870-2.

Mustoe, Anthony M. et al. (Mar. 2018). "Pervasive Regulatory Functions of mRNA Structure Revealed by High-Resolution SHAPE Probing". In: *Cell* 173.1, 181–195.e18. ISSN: 0092-8674. DOI: 10.1016/j.cell.2018.02.034.

Nam, Jin-Wu, Seo-Won Choi, and Bo-Hyun You (May 2016). "Incredible RNA: Dual Functions of Coding and Noncoding". In: *Molecules and Cells* 39.5, pp. 367–374. ISSN: 1016-8478. DOI: 10.14348/molcells.2016.0039.

Negri, Tatianne da Costa et al. (Apr. 2018). "Pattern recognition analysis on long noncoding RNAs: a tool for prediction in plants". In: *Briefings in Bioinformatics* 20.2, pp. 682–689. ISSN: 1477-4054. DOI: 10.1093/bib/bby034.

Nemeth, Kinga et al. (Nov. 2023). "Non-coding RNAs in disease: from mechanisms to therapeutics". In: *Nature Reviews Genetics* 25.3, pp. 211–232. ISSN: 1471-0064. DOI: 10.1038/s41576-023-00662-1.

Nguyen, Eric et al. (2023). *HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution*. DOI: 10.48550/ARXIV.2306.15794.

O'Leary, Nuala A. et al. (Nov. 2015). "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation". In: *Nucleic Acids Research* 44.D1, pp. D733–D745. ISSN: 1362-4962. DOI: 10.1093/nar/gkv1189.

Pang, Yanan, Chuanbin Mao, and Shanrong Liu (2018). "Encoding activities of non-coding RNAs". In: *Theranostics* 8.9, pp. 2496–2507. ISSN: 1838-7640. DOI: 10.7150/thno.24677.

Penić, Rafael Josip et al. (2024). *RiNALMo: General-Purpose RNA Language Models Can Generalize Well on Structure Prediction Tasks*. DOI: 10.48550/ARXIV.2403.00043.

Pian, Cong et al. (May 2016). "LncRNApred: Classification of Long Non-Coding RNAs and Protein-Coding Transcripts by the Ensemble Algorithm with a New Hybrid Feature". In: *PLOS ONE* 11.5. Ed. by Vinod Scaria, e0154567. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0154567.

Press, Ofir, Noah A. Smith, and Mike Lewis (2021). *Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation*. DOI: 10.48550/ARXIV.2108.12409.

Romeijn, Luuk, Andrius Bernatavicius, and Duong Vu (Aug. 2024). "MycoAI: Fast and accurate taxonomic classification for fungal ITS sequences". In: *Molecular Ecology Resources*. ISSN: 1755-0998. DOI: 10.1111/1755-0998.14006.

Romeijn, Luuk, Davy Cats, et al. (Jan. 2025). "LncRNA-BERT: An RNA Language Model for Classifying Coding and Long Non-Coding RNA". In: DOI: 10.1101/2025.01.09.632168.

Sanabria, Melissa et al. (July 2024). "DNA language model GROVER learns sequence context in the human genome". In: *Nature Machine Intelligence* 6.8, pp. 911–923. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00872-0.

Schneider, Hugo W. et al. (Oct. 2017). "A Support Vector Machine based method to distinguish long non-coding RNAs from protein coding transcripts". In: *BMC Genomics* 18.1. ISSN: 1471-2164. DOI: 10.1186/s12864-017-4178-4.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. DOI: 10.18653/v1/P16-1162.

Simopoulos, Caitlin M. A., Elizabeth A. Weretilnyk, and G. Brian Golding (May 2018). "Prediction of plant lncRNA by ensemble machine learning classifiers". In: *BMC Genomics* 19.1. ISSN: 1471-2164. DOI: 10.1186/s12864-018-4665-2.

Singh, Dalwinder and Joy Roy (Nov. 2022). "A large-scale benchmark study of tools for the classification of protein-coding and non-coding RNAs". In: *Nucleic Acids Research* 50.21, pp. 12094–12111. ISSN: 1362-4962. DOI: 10.1093/nar/gkac1092.

Singh, Urminder et al. (Oct. 2017). "PLncPRO for prediction of long non-coding RNAs (lncRNAs) in plants and its application for discovery of abiotic stress-responsive lncRNAs in rice and chickpea". In: *Nucleic Acids Research* 45.22, e183–e183. ISSN: 1362-4962. DOI: 10.1093/nar/gkx866.

Sloma, Michael F. and David H. Mathews (Oct. 2016). "Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures". In: *RNA* 22.12, pp. 1808–1818. ISSN: 1469-9001. DOI: 10.1261/rna.053694.115.

Su, Jianlin et al. (2021). *RoFormer: Enhanced Transformer with Rotary Position Embedding*. DOI: 10.48550/ARXIV.2104.09864.

Sun, Kun et al. (Feb. 2013). "iSeeRNA: identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data". In: *BMC Genomics* 14.S2. ISSN: 1471-2164. DOI: 10.1186/1471-2164-14-s2-s7.

Sun, Lei et al. (Oct. 2015). "lncRScan-SVM: A Tool for Predicting Long Non-Coding RNAs Using Support Vector Machine". In: *PLOS ONE* 10.10. Ed. by Gajendra P. S. Raghava, e0139654. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0139654.

Sun, Liang et al. (July 2013). "Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts". In: *Nucleic Acids Research* 41.17, e166–e166. ISSN: 0305-1048. DOI: 10.1093/nar/gkt646.

Sweeney, Blake A et al. (Oct. 2020). "RNAcentral 2021: secondary structure integration, improved sequence search and new member databases". In: *Nucleic Acids Research* 49.D1, pp. D212–D220. ISSN: 1362-4962. DOI: 10.1093/nar/gkaa921.

Tahmid, Md Toki et al. (July 2024). *BiRNA-BERT allows efficient RNA language modeling with adaptive tokenization*. DOI: 10.1101/2024.07.02.601703.

Tong, Xiaoxue and Shiyong Liu (Feb. 2019). "CPPred: coding potential prediction based on the global description of RNA sequence". In: *Nucleic Acids Research* 47.8, e43–e43. ISSN: 1362-4962. DOI: 10.1093/nar/gkz087.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: 1706.03762.

Wang, Guangyu et al. (Jan. 2019). "Characterization and identification of long non-coding RNAs based on feature relationship". In: *Bioinformatics* 35.17. Ed. by Alfonso Valencia, pp. 2949–2956. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btz008.

Wang, Liguo et al. (Jan. 2013). "CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model". In: *Nucleic Acids Research* 41.6, e74–e74. ISSN: 0305-1048. DOI: 10.1093/nar/gkt006.

Wang, Ning et al. (May 2024). "Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning". In: *Nature Machine Intelligence* 6.5, pp. 548–557. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00836-4.

Wang, Xi et al. (July 2023). *UNI-RNA: UNIVERSAL PRE-TRAINED MODELS REVOLUTIONIZE RNA RESEARCH*. DOI: 10.1101/2023.07.11.548588.

Wang, Ying et al. (Apr. 2023). "LncDLSM: Identification of Long Non-Coding RNAs With Deep Learning-Based Sequence Model". In: *IEEE Journal of Biomedical and Health Informatics* 27.4, pp. 2117–2127. ISSN: 2168-2208. DOI: 10.1109/jbhi.2023.3247805.

Weikard, Rosemarie, Frieder Hadlich, and Christa Kuehn (2013). "Identification of novel transcripts and noncoding RNAs in bovine skin by deep next generation sequencing". In: *BMC Genomics* 14.1, p. 789. ISSN: 1471-2164. DOI: 10.1186/1471-2164-14-789.

Wucher, Valentin et al. (Jan. 2017). "FEELnc: a tool for long non-coding RNA annotation and its application to the dog transcriptome". In: *Nucleic Acids Research*, gkw1306. ISSN: 1362-4962. DOI: 10.1093/nar/gkw1306.

Yang, Cheng et al. (May 2018). "LncADeep: anab initiolncRNA identification and functional annotation tool based on deep learning". In: *Bioinformatics* 34.22. Ed. by Inanc Birol, pp. 3825–3834. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bty428.

Yang, Meng et al. (May 2022). "Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution". In: *Nucleic Acids Research* 50.14, e81–e81. ISSN: 1362-4962. DOI: 10.1093/nar/gkac326.

Yang, Sen et al. (Feb. 2020). "NCResNet: Noncoding Ribonucleic Acid Prediction Based on a Deep Resident Network of Ribonucleic Acid Sequences". In: *Frontiers in Genetics* 11. ISSN: 1664-8021. DOI: 10.3389/fgene.2020.00090.

Yin, Weijie et al. (Mar. 2024). *ERNIE-RNA: An RNA Language Model with Structure-enhanced Representations*. DOI: 10.1101/2024.03.17.585376.

Zaheer, Manzil et al. (2020). "Big Bird: Transformers for Longer Sequences". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 17283–17297.

Zhang, Yikun et al. (Nov. 2023). "Multiple sequence alignment-based RNA language model and its application to structural inference". In: *Nucleic Acids Research* 52.1, e3–e3. ISSN: 1362-4962. DOI: 10.1093/nar/gkad1031.

Zhang, Yu et al. (Mar. 2020). "DeepCPP: a deep neural network based on nucleotide bias information and minimum distribution similarity feature selection for RNA coding potential prediction". In: *Briefings in Bioinformatics* 22.2, pp. 2073–2084. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa039.

Zhao, Jian, Xiaofeng Song, and Kai Wang (Oct. 2016). "lncScore: alignment-free identification of long noncoding RNA from assembled novel transcripts". In: *Scientific Reports* 6.1. ISSN: 2045-2322. DOI: 10.1038/srep34838.

Zhao, Lianhe et al. (Nov. 2020). "NONCODEV6: an updated database dedicated to long non-coding RNA annotation in both animals and plants". In: *Nucleic Acids Research* 49.D1, pp. D165–D171. ISSN: 1362-4962. DOI: 10.1093/nar/gkaa1046.

Zheng, Hansi et al. (Aug. 2021). "A systematic evaluation of the computational tools for lncRNA identification". In: *Briefings in Bioinformatics* 22.6. ISSN: 1477-4054. DOI: 10.1093/bib/bbab285.

Zhou, Jian and Olga G Troyanskaya (Aug. 2015). "Predicting effects of noncoding variants with deep learning–based sequence model". In: *Nature Methods* 12.10, pp. 931–934. ISSN: 1548-7105. DOI: 10.1038/nmeth.3547.

Zhou, Zhihan, Yanrong Ji, et al. (2023). *DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genome*. DOI: 10.48550/ARXIV.2306.15006.

Zhou, Zhihan, Weimin Wu, et al. (2024). *DNABERT-S: Pioneering Species Differentiation with Species-Aware DNA Embeddings*. DOI: 10.48550/ARXIV.2402.08777.

Appendix A

Supplementary Information on Feature-Based Approach

A.1 List of Features Included in Analysis

Table A.3 describes all 8610 features extracted from the GENCODE training set by the lncRNA-Py package that were used in the EDA and feature selection procedure (Section 3.2, 4.2). The majority of features correspond to frequencies of specific k-mers. Some features in the 'Secondary Structure' category based on the ViennaRNA package were left out from the analysis as they could not be calculated within 7 days on SHARK, even though they are implemented in the lncRNA-Py package (`lncrnapy.features.sse`). Note that lncRNA-Py does not contain feature extractor classes for features based on genome mapping, as this would require a GTF input format and cannot be applied to novel transcripts directly.

A.2 LncRNA-LR and LncRNA-RF Feature Importance

The 25 most important features of lncRNA-LR and lncRNA-RF are listed in Table A.2. Importance is evaluated using the coefficient size for logistic regression and Gini importance for the random forest. Relative RFE (Section 3.2, Algorithm 1) is applied to select 10 and 100 features (resp.) for lncRNA-LR and lncRNA-RF.

A.3 Validating the CPAT Re-Implementation

To validate the reliability of the lncRNA-Py package, we compare our re-implementation of the CPAT algorithm to the results presented in (L. Wang et al. 2013) and official software (`https://sourceforge.net/projects/rna-cpat/`), which should show similar behaviour. The results in Figure A.1 and Table A.1 indicate a discrepancy between the results presented in the publication, the current software release, and the re-implementation by lncRNA-Py. We identify the following possible reasons for these differences, which generally apply to all features and algorithms that lncRNA-Py contains:

1. The work does not specify all details about the feature or implemented methods.
2. The specified methods deviates from the implemented methods in the code.

CPAT variant	**F1 (macro)**	**Precision**		**Recall**	
		pcRNA	**ncRNA**	**pcRNA**	**ncRNA**
lncRNA-Py	0.95	0.96	0.95	0.95	0.96
Paper	0.97	0.97	0.96	0.96	0.97
Software	0.96	0.95	0.98	0.98	0.95

Table A.1: Macro-averaged F1-score, precision, and recall for the re-implementation of CPAT in the lncRNA-Py package, for the results presented in its publication (L. Wang et al. 2013), and for the predictions obtained using the official software.

	Name	Extractor class	Description
-	Length	`Length`	Transcript length
1	ORF	`ORFCoordinates`	ORF coordinates, for 5 relaxation levels (Wucher et al. 2017)
	ORF length	`ORFLength`	ORF length, for 5 relaxation levels
	ORF coverage	`ORFCoverage`	ORF / sequence length, for 5 relaxation levels
	UTR length	`UTRCoordinates`	5' UTR and 3' UTR length
	UTR coverage	`UTRLength`	UTR / sequence length
2	Fickett score	`FickettScore`	(Fickett 1982; L. Wang et al. 2013)
	Complexity	`Complexity`	Local compositional complexity (entropy)
	ORF amino acid frequency	`KmerFreqs`	-
	GC content	`GCContent`	Proportion of C/G in sequence
	Nucleotide distribution	`SequenceDistribution`	Nucleotide proportion (in sequence/ORF) for every 1/4 transcript length
	SCS	`StdStopCodons`	Standard deviation of stop codon counts between reading frames (Lei Sun et al. 2015)
	1-mer frequency	`KmerFreqs`	In sequence & ORF
	2-mer frequency	`KmerFreqs`	In sequence & ORF
	2-mer EDP (ORF)	`EntropyDensityProfile`	2-mer entropy density profile of ORF
	1-gapped 2-mer frequency	`KmerFreqs`	Discontinuous k-mers (Yu Zhang et al. 2020)
	2-gapped 2-mer frequency	`KmerFreqs`	Discontinuous k-mers
	3-mer frequency	`KmerFreqs`	In sequence & ORF
	3-mer frequency (PLEK)	`KmerFreqs`	PLEK-corrected frequencies (A. Li et al. 2014)
	3-mer entropy	`Entropy`	3-mer frequency entropy, in sequence & ORF
	6-mer frequency	`KmerFreqs`	In sequence & ORF
	6-mer score	`KmerScore`	Hexamer bias (L. Wang et al. 2013)
	Zhang score	`ZhangScore`	Nucleotide bias around start codon (Yu Zhang et al. 2020)
	MLCDS	`MLCDS`	6 Most-Like Coding Sequence (+ scores) (Liang Sun et al. 2013)
	MLCDS length	`MLCDSLength`	Length of top 6 MLCDS (Liang Sun et al. 2013)
	MLCDS length-percentage	`MLCDSLengthPercentage`	Length of top MLCDS / lengths of other 5 MLCDSs
	MLCDS length std	`MLCDSLengthStd`	Standard deviation between top 6 MLCDS lengths (Guo et al. 2019)
	MLCDS score-distance	`MLCDSScoreDistance`	Sum of difference between top MLCDS score and others
	MLCDS score std	`MLCDSScoreStd`	Standard deviation between top 6 MLCDS scores (Guo et al. 2019)
	6-mer ORF distance (ratio)	`KmerDistance`	Euclidian distance (ratio) of 6-mer ORF frequency to average pcRNA/ncRNA spectrum (Han, Liang, Ma, et al. 2018)
3	BLASTX hits	`BLASTXSearch`	Number of BLASTX hits in UniRef90 protein database (Kong et al. 2007)
	BLASTX hit score	`BLASTXSearch`	Mean of mean log e-value over three reading frames
	BLASTX frame score	`BLASTXSearch`	Mean of deviation from BLASTX hit score
	BLASTX S-score	`BLASTXSearch`	Sum of logs of significant scores (U. Singh et al. 2017)
	BLASTX bit score	`BLASTXSearch`	Total bit score
	BLASTX frame entropy	`BLASTXSearch`	Entropy of probability of hits in i-th reading frame
	BLASTX identity	`BLASTXSearch`	Sum of the identity percentage
	BLASTX hits >0	`BLASTXBinary`	Binary indicator of whether or not a hit is found (suggested by H. Mei)
4	ORF pI	`ORFProteinAnalysis`	Isoelectric point of ORF-encoded protein
	ORF MW	`ORFProteinAnalysis`	Molecular weight of ORF-encoded protein
	ORF aromaticity	`ORFProteinAnalysis`	Aromaticity of ORF-encoded protein
	ORF instability	`ORFProteinAnalysis`	Instability index of ORF-encoded protein
	ORF gravy	`ORFProteinAnalysis`	Gravy of ORF-encoded protein
	EIIP 1/3	`EIIPPhysicoChemical`	Power at 1/3 of Electron-Ion Interaction Profile power spectrum (Han, Liang, Ma, et al. 2018)
	EIIP SNR	`EIIPPhysicoChemical`	Signal-to-noise ratio of EIIP 1/3
	EIIP Q1	`EIIPPhysicoChemical`	First quantile of EIIP power spectrum
	EIIP Q2	`EIIPPhysicoChemical`	Second quantile of EIIP power spectrum
	EIIP min/max	`EIIPPhysicoChemical`	Of EIIP power spectrum
6	ORF helix	`ORFProteinAnalysis`	Fraction of amino acids that tend to be helix
	ORF turn	`ORFProteinAnalysis`	Fraction of amino acids that tend to be turn
	ORF sheet	`ORFProteinAnalysis`	Fraction of amino acids that tend to be sheet

Table A.2: Description of the 8610 features extracted from the GENCODE dataset used in the exploratory data analysis and feature selection procedure of lncRNA-LR and lncRNA-RF. The first column refers to feature types: 1) ORF; 2) sequence patterns; 3) database alignment; 4) genome mapping (not implemented and excluded); 5) physicochemical; 6) secondary structure. Extractor classes are part of the `features` module of `lncrnapy`. Many of the features listed here are used by multiple lncRNA classification algorithms, works that introduced the feature or are characterized by it are cited.

Rank	Logistic Regression (lncRNA-LR)		Random Forest (lncRNA-RF)	
	Feature	**Coefficient**	**Feature**	**Gini importance**
1	ORF1 length	21.978	BLASTX hit score	0.074
2	Y (ORF,aa)	-13.080	ORF3 length	0.070
3	ORF length	-11.199	BLASTX S-score	0.066
4	TAT (ORF) s=3	9.112	ORF1 length	0.058
5	TAC (ORF) s=3	8.513	BLASTX bit score	0.048
6	EIIP 1_3	-7.532	ORF2 length	0.041
7	TGG (ORF) s=3	-6.111	ORF4 length	0.041
8	W (ORF,aa)	5.998	BLASTX identity	0.033
9	EIIP SNR	3.777	BLASTX hits >0	0.032
10	ORF MW	0.944	ORF MW	0.031
11			BLASTX hits	0.029
12			ORF length	0.029
13			BLASTX frame entropy	0.027
14			ORF 6-mer eucDistRatio s=3	0.022
15			ORF3 coverage	0.021
16			ORF 6-mer eucDist pc s=3	0.017
17			ORF1 coverage	0.017
18			ORF 6-mer eucDist nc s=3	0.016
19			ORF4 coverage	0.016
20			Fickett score	0.014
21			EIIP SNR	0.013
22			EIIP 1_3	0.012
23			BLASTX frame score	0.011
24			3-mer ORF entropy	0.010
25			MLCDS5 score	0.009

Table A.3: The 10 and 25 most important features of the feature-based lncRNA-LR (Logistic Regression) and lncRNA-RF (Random Forest) models, fit on the GENCODE training dataset. Features based on BLAST, ORF, and sequence organization (EIIP, 6-mer distance) are considered as most important. 10 and 100 features (out of 8610) were selected for lncRNA-LR and -RF respectively using relative Recursive Feature Elimination (Section 3.2, Algorithm 1).

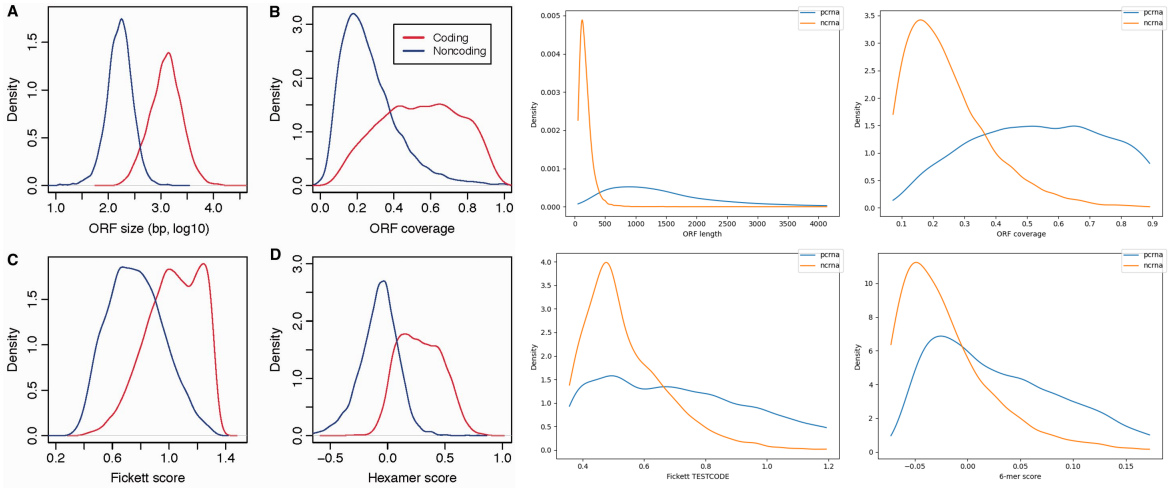

Figure A.1: Density plots for CPAT's four basic features as presented in their paper (left) (L. Wang et al. 2013, Figure 1)) and by our own re-implemented versions (right).

3. The data may be preprocessed in a way that is not specified in the paper.

4. The feature is re-implemented wrongly.

The implementation of the Fickett score feature was verified with an online tool `https://gcat.davidson.edu/DGPB/testcode.html`, so we are confident that the last reason does not apply here. Reason 2 may be caused by software updates over time. For example, inspecting CPAT's source code reveals that its current implementation identifies multiple ORF candidates and evaluates the Fickett score for each of them. This contrasts the description given in (L. Wang et al. 2013), which states that 'The Fickett score is independent of the ORF'. The performance obtained by lncRNA-Py's variant of CPAT is comparable yet slighter lower (Table A.1) than the scores reported in the original paper.

For the above listed reasons, we anticipate that deviations between re-implemented features and algorithms and those in official software releases are likely to occur for any lncRNA classifciation algorithm. For the sake of credibility, we thus choose to only report performances of officially released software in our results.

```
python -m experiments.cpat_validation
```

Appendix B

LncRNA Classifier Comparison Table

The values displayed in Figure 4.1 are listed in Table B.1.

Test set	Metrics	lncRNA-BERT		lncRNA-LR	lncRNA-RF	CPAT	LncFinder	PredLnc	LncADeep	RNAsamba	mRNN
		K-mer (k=3)	CSE (k=9)								
GENCODE/ RefSeq	F1 (macro)	0.940	0.943	0.861	0.953	0.889	0.866	0.941	0.930	0.902	0.907
	Precision (pcRNA)	0.965	0.957	0.967	0.960	0.943	0.962	0.956	0.966	0.947	0.949
	Recall (pcRNA)	0.958	0.971	0.838	0.981	0.910	0.851	0.969	0.942	0.924	0.929
	Precision (ncRNA)	0.912	0.937	0.733	0.959	0.824	0.748	0.934	0.884	0.848	0.858
	Recall (ncRNA)	0.926	0.907	0.940	0.914	0.884	0.930	0.906	0.929	0.892	0.895
	Accuracy	0.947	0.950	0.871	0.960	0.902	0.876	0.949	0.938	0.914	0.918
CPAT	F1 (macro)	0.963	0.947	0.939	0.973	0.962	0.959	0.950	0.970	0.958	0.950
	Precision (pcRNA)	0.942	0.914	0.957	0.955	0.948	0.959	0.917	0.958	0.933	0.937
	Recall (pcRNA)	0.987	0.988	0.920	0.992	0.977	0.960	0.990	0.983	0.988	0.965
	Precision (ncRNA)	0.986	0.987	0.923	0.992	0.977	0.960	0.989	0.983	0.987	0.963
	Recall (ncRNA)	0.940	0.906	0.959	0.954	0.947	0.959	0.910	0.957	0.928	0.935
	Accuracy	0.963	0.947	0.939	0.973	0.962	0.959	0.950	0.970	0.958	0.950
RNAChallenge	F1 (macro)	0.235	0.242	0.037	0.316	0.031	0.005	0.173	0.146	0.138	0.155
	Precision (pcRNA)	0.397	0.412	0.048	0.402	0.069	0.006	0.337	0.288	0.278	0.295
	Recall (pcRNA)	0.420	0.453	0.033	0.301	0.050	0.004	0.344	0.272	0.259	0.278
	Precision (ncRNA)	0.066	0.056	0.030	0.248	0.003	0.005	0.005	0.012	0.008	0.022
	Recall (ncRNA)	0.060	0.048	0.044	0.340	0.004	0.007	0.005	0.013	0.008	0.024
	Accuracy	0.274	0.289	0.037	0.317	0.031	0.005	0.207	0.167	0.158	0.175

Table B.1: Performance of lncRNA classifiers on three test sets. Values correspond to the bar charts displayed in Figure 4.1.

Appendix C

Hyperparameter Tuning

A description of our hyperparameter tuning procedure is provided on our GitHub, specifically at `https://luukromeijn.github.io/lncRNA-Py/experiments.html#hyperparameter-tuning`.