Politecnico di Torino

Politecnico di Torino

MSc. Data Science and Engineering

A.a. 2024/2025

Graduation Session October 2025

Heuristic Algorithm for Predicting Alternatively Spliced mRNAs with Pre-Trained LLM in Cancer

Supervisors:

Stefano Di Carlo
Roberta Bardini
Alessandro Savino
Matteo Cereda
Lorenzo Martini

Candidate:

Gustavo
Nicoletti Rosa

Abstract

Alternative Splicing is the RNA's ability to be spliced into many different mRNA isoforms, and it is of great evolutionary importance because it allows a single gene to produce a variety of proteins. However, in cancer, the spliceosome machinery produces aberrant isoforms or changes their expression, which alters the behavior of the cell, as they interfere with biological pathways.

The study of novel cancer isoforms is essential for developing therapies that can suppress their expression or exploit the new epitopes, in addition to providing a deeper understanding of the disease.

The relatively new long-read sequencing technology enables a more accurate representation of the transcriptome than the older short-read. Still, not all isoforms have been sequenced, and each cell in each state will produce different outcomes. Therefore, a way of predicting possible isoforms is an interesting problem. As we see in this thesis, generating all possible isoforms solely based on the main splicing signals of the genome takes virtually infinite time and mostly inaccurate results.

Hence, we propose a heuristic algorithm for the prediction of tumoral isoforms with the inclusion of a Large Language Model pre-trained on RNA long-reads of multiple tumoral cell lines. We evaluate our algorithm by analysing its perplexity, computation time, and comparing our results with a prostate cancer long-read dataset provided by the Italian Institute for Genomic Medicine (IIGM).

Acknowledgements

I would like to thank all my supervisors and collaborators during this thesis for the opportunity and trust to study such a diverse and interesting topic, in this initiative to cooperate between the Polytechnic of Turin and the Italian Institute of Medical Genomics. Professor Roberta Bardini has been of great support both academically, with important insights on how to improve my work, and also very kind in moments of struggle. Another special gratitude goes to the PhD researchers Lorenzo Martini and Tommaso Becchi, for the knowledge you have shared with me and the ideas we discussed.

I appreciate the support my family has given me, especially my mom. My growth was only possible with yours. We have come a long way from the crazy streets of Brás (São Paulo, Brazil); this degree is a fruit of your sacrifices, too.

To my friends from Collegio Einaudi, I will never forget the moments we have shared in these 5 years; having you all in my life has been a blessing, for you have shaped the person I am now in many ways. And the incredible IEEE-HKN student society, the number of opportunities you have given me is really amazing, initially from the passion of teaching Signal Analysis to students, to taking a role on the Board of administration, going to a conference in America, and finally being Advisor for two new Boards, all of it, with friends whom I can trust and enjoy a good time together!

Table of Contents

List of Tables IV

List of Figures V

Glossary VII

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 1
- 1.3 Objectives and Scope of the Study . . . 2
- 1.4 Thesis Overview . . . 2

2 Background 4
- 2.1 Transcriptomics . . . 4
  - 2.1.1 Sequencing . . . 4
  - 2.1.2 From genetic code to proteins . . . 6
  - 2.1.3 Splicing . . . 7
  - 2.1.4 Alternative Splicing (AS) . . . 9
  - 2.1.5 Alternative Splicing in Cancer . . . 11
  - 2.1.6 Clinical implications . . . 12
- 2.2 Deep Learning and Large Language Models . . . 13
- 2.3 Deep Learning in Genomics . . . 16
  - 2.3.1 Bidirectional Encoder Representations from Transformers model for DNA (DNABERT) . . . 16
  - 2.3.2 HyenaDNA . . . 17
  - 2.3.3 BigRNA . . . 17
  - 2.3.4 SpliceBERT . . . 17
  - 2.3.5 Long read RNA with Striped Hyena (LoRNA^{SH}) . . . 18
  - 2.3.6 Evo 2 . . . 19
  - 2.3.7 Our use case . . . 19

3 Materials and Methods 21
3.1 Datasets . . . 21
3.1.1 IIGM-dataset . . . 21
3.1.2 Genome Browser . . . 21
3.2 Generation of isoform with splicing signals . . . 23
3.2.1 Basic splicing signal motifs . . . 24
3.3 LoRNA^{SH} Pure Generations . . . 28
3.3.1 Introns and Exons Statistics . . . 29
3.3.2 SpliceAI . . . 30
3.4 Greedy algorithm . . . 30
3.4.1 The algorithm . . . 30
3.4.2 Perplexity of the algorithm . . . 32
3.4.3 Counting the created isoforms . . . 33
3.4.4 Distribution of Exon expression . . . 34
3.4.5 Computational analysis . . . 34
3.5 Base-level Binary Classifier . . . 34
3.5.1 Validation . . . 34

4 Results 37
4.1 Generation of isoform with splicing signals . . . 37
4.2 Reproducing LoRNA^{SH} Results . . . 38
4.2.1 BLAT . . . 39
4.2.2 Introns and Exons Statistics . . . 40
4.2.3 SpliceAI on generated Splicing Sites . . . 42
4.2.4 Expression vs Full sequence and End token probabilities . . . 43
4.3 Greedy algorithm . . . 45
4.3.1 Exon expression in Greedy generations . . . 52
4.3.2 Perplexity . . . 63
4.4 Base-level Binary Classifier . . . 65
4.4.1 ROC curves and AUC . . . 65
4.4.2 Final test . . . 69
4.5 Hardware resources in Greedy algorithm . . . 72
4.5.1 Algorithm time complexity . . . 72
4.5.2 GPU memory usage . . . 74

5 Conclusion 76
5.1 Future works . . . 77

Bibliography 78

List of Tables

2.1 Comparison of sequencing technologies 7

3.1 Donor position weight matrix 25
3.2 Acceptor position weight matrix 26
3.3 Top 10 Donor motifs 27
3.4 Top 10 Acceptor motifs 28
3.5 Interpretation of AUC values 36

4.1 Information of two sample genes 38
4.2 List of ranks of observed motifs in FUS gene 38
4.3 Sample Sizes of Pure Generations and IIGM Statistics 42
4.4 Information on 21 isoforms 45
4.5 Average AUC across 19 isoforms for multiple genes and ranges . . . 69
4.6 Long-expression weighted average AUC across 19 isoforms for multiple genes and ranges 69
4.7 GPU configurations 74

List of Figures

2.1 Alternative Splicing from DNA to mRNAs 8
2.2 Alternative Splicing mechanisms 10

3.1 Gene length distribution on IIGM's dataset 22
3.2 Chromosome distribution on IIGM's dataset 22
3.3 Genome Browser example 23
3.4 Genome Browser example - Zoom 23
3.5 Distribution of 3'SS and 5'SS motif probabilities 27
3.6 Distance distribution between branch and acceptor splice signals . . 28
3.7 Greedy Algorithm 32
3.8 Final step of the greedy algorithm 33
3.9 Mesh for counting generated isoforms 34
3.10 Exon classifier example 35
3.11 ROC curve example 36

4.1 BLAT results from pure generations 39
4.2 Distribution of pure generations statistics from original paper . . . 40
4.3 Distribution of exon and intron counters in pure generations and reference 41
4.4 Distribution of exon, intron, sequence, and mRNA lengths in pure generations and reference 41
4.5 Distribution of SpliceAI probabilities in pure generations and reference 43
4.6 Model probabilities and Dataset Expression 44
4.7 Examples of greedy-generated isoforms in UCSC - HRAS 46
4.8 Top 10 greedy and IIGM isoforms - FOXA1 47
4.9 Top 10 greedy and IIGM isoforms - HRAS 48
4.10 Top 10 greedy and IIGM isoforms - MYC 49
4.11 Top 10 greedy and IIGM isoforms - KLF6 50
4.12 Top 10 greedy and IIGM isoforms - MDM2 51
4.13 Top 10 greedy and IIGM isoforms - SRSF1 52
4.14 Exon expression of greedy generations - FOXA1 53

4.15 Exon expression of greedy generations - HRAS 54
4.16 Exon expression of greedy generations - MYC 55
4.17 Exon expression of greedy generations - KLF6 56
4.18 Exon expression of greedy generations - MDM2 57
4.19 Exon expression of greedy generations - SRSF1 58
4.20 Correlation of Exon/Intron expression - FOXA1 59
4.21 Correlation of Exon/Intron expression - HRAS 59
4.22 Correlation of Exon/Intron expression - MYC 59
4.23 Correlation of Exon/Intron expression - KLF6 60
4.24 Correlation of Exon/Intron expression - MDM2 60
4.25 Correlation of Exon/Intron expression - SRSF1 60
4.26 Perplexity of all decisions made by the Greedy algorithm 64
4.27 Perplexity of decisions that change state made by the Greedy algorithm 64
4.28 Perplexity of all decisions made by the Greedy algorithm by gene . 64
4.29 Perplexity of decisions that change state made by the Greedy algorithm by gene 65
4.30 ROC curve example - FOXA1 66
4.31 ROC curve example - HRAS 67
4.32 ROC curve example - MYC 67
4.33 ROC curve example - KLF6 68
4.34 ROC curve example - SRSF1 68
4.35 WAUC distribution on final test by genes 70
4.36 AUC distribution on final test by isoform 71
4.37 AUC distribution on final test by isoform type 71
4.38 AUC vs log(Long-read expression) 72
4.39 Execution, GPU, and CPU times in Greedy algorithm 73
4.40 GPU limits on IIGM's dataset 75
4.41 Peak VRAM usage during Greedy algorithm 75

Glossary

3'SS	3' Splice Site.
5'SS	5' Splice Site.
A	Adenine.
Ad2	Adenovirus type 2.
ANN	Artificial Neural Network.
API	Application Programming Interface.
AS	Alternative SPlicing.
AUC	Area Under Curve.
BBP	Branch Point Binding Protein.
BLAST	Basic Local Alignment Search Tool.
BLAT	BLAST-like alignment tool.
BP	Branching Point.
C	Cytosine.
CART-T	Chimeric Antigen Receptor T cell.
cDNA	Complementary DNA.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
ddNTP	Dideoxynucleotide Triphosphate.
DL	Deep Learning.
DNA	Deoxyribonucleic Acid.
DNABERT	Bidirectional Encoder Representations from Transformers from for DNA.
dNTP	Deoxynucleotide Triphosphate.
dRNA	Direct RNA Sequencing.
ESE	Exonic Splicing Enhancer.
ESS	Exonic Splicing Silencers.

FFT	Fast Fourier Transform.
FN	False Negative.
FOGSAA	Fast Optimal Global Alignment Algorithm.
FOXA1	Forkhead box protein A1.
FP	False Positive.
G	Guanine.
GENCODE	Genome Encyclopedia of DNA Elements.
GPU	Graphics Processing Unit.
GRCh37	Genome Reference Consortium Human build 37.
HAVANA	Human and Vertebrate Analysis and Annotation.
HGP	Human Genome Project.
HRAS	Harvey Rat Sarcoma proto-oncogene.
HSP	High Scoring Pairs.
IIGM	Italian Institute of Medical Genomics.
ISE	Intronic Splicing Enhancers.
ISS	Intronic Splicing Silencers.
KLF6	Krueppel-like factor 6.
LLM	Large Language Model.
LoRNA^{SH}	Long Read RNA with Striped Hyena.
LSTM	Long Short-Term Memory.
MDM2	Mouse double minute 2 homolog.
MDS	Myelodysplastic Syndromes.
MHA	Multi-Head Attention.
miRNA	Micro RNA.
ML	Machine Learning.
mRNA	Messenger Ribonucleic Acid.
MYC	Myelocytomatosis family proto-oncogene.
NGS	Next Generation Sequencing.
NLP	Natural Language Processing.
NMD	Nonsense-Mediated Decay.
NTP	Next Token Prediction.

ONT	Oxford Nanopore Technologies.
ORF	Open Reading Frame.
PAS	Polyadenylation Site.
PC3	Human Prostatic Carcinoma Cell Line.
PCC	Pearson Correlation Coefficient.
PCR	Polymerase Chain Reaction.
PHAST	Phylogenetic Analysis with Space/Time models.
phylo-HMM	Phylogenetic Hidden Markov Chain.
PMC	Premature Termination Codon.
PPT	Polypyrimidine Tract.
pre-mRNA	precursor messenger RNA.
PWM	Position Weight Matrix.
RBP	RNA-Binding Protein.
REST	Representational State Transfer.
RNA	Ribonucleic Acid.
RNN	Recurrent Neural Network.
RNP	Ribonucleoprotein.
ROC	Receiver Operating Characteristic.
rRNA	Ribosomal RNA.
SBS	Sequencing by Synthesis.
SF1	Splicing Factor 1.
snoRNA	Small Nucleolar RNA.
snRNA	Small Nuclear RNA.
SRSF1	Serine/arginine-rich splicing factor 1.
T	Thymine.
TMB	Tumor Mutation Burden.
TME	Tumor Microenvironment.
TN	True Negative.
TP	True Positive.
TPM	Transcripts per Million.
tRNA	Transport RNA.
TSS	Transcription Start Site.
U	Uracil.
U2AF	U2 Auxilary Factor.

UCSC University of California Santa Cruz.
UTR Untranslated Region.

VRAM Video Random Access Memory.

Chapter 1

Introduction

1.1 Motivation

According to global cancer statistics of 2022, cancer is responsible for 16.8% of global deaths, and leading premature deaths from noncommunicable diseases in 177 of 183 countries, in those aged $30 - 69$ years, at 30.3% [1]. A key factor for understanding cancer lies in which proteins are expressed and how they are involved in the development of the disease. The proteins are produced in the ribosomes by translating mRNA transcripts, which are derived from the splicing of pre-mRNA inside the nucleus of the cell. As we will see in this thesis, the splicing mechanism is complex and non-deterministic, giving rise to the concept of Alternative Splicing (AS), increasing the challenge for researchers, since a single gene can produce several mRNA isoforms. Mutations in the genome can cause aberrant alternative splicing, creating proteins that disrupt the natural biological pathways of the cell, e.g., suppressing apoptosis [2], and with such knowledge, scientists are able to propose new therapies for patients [3]. Considering that dysregulated RNA splicing characterizes almost all types of cancer, identifying the isoforms in cancer cells is of high interest for the cancer research field [4].

1.2 Problem Statement

Current sequencing technology, although very powerful, cannot read every mRNA isoform inside a replicate, especially the ones with lower expression, simply due to probabilistic factors. But these unsequenced isoforms may be of interest when studying cancer. The problem we aim to solve is the prediction of isoforms in cancer through computational methods with one of the most recent state-of-the-art Large Language Model (LLM) architectures.

1.3 Objectives and Scope of the Study

We will study the feasibility of a combinatorial approach in the production of isoforms by considering the most commonly used splicing site motifs. This will be followed by the creation of a heuristic algorithm able to exploit LLM's capabilities for solving our task. We will explore the results of our algorithm by comparing them with the GRCh37 genome reference and a dataset of prostate cancer long-reads by the Italian Institute of Genomic Medicine (IIGM), with whom we collaborated in the development of this work. As well as making considerations on the connection between our outputs and the conservation of bases across a hundred vertebrates [5, 6]. The algorithm will first be analyzed both discoursively and with standard metrics in a small set of genes, then a general test will follow with more than 50 genes to gather a better statistic of its performance. Lastly, a short analysis of the hardware resources is presented, to enable users to make the correct considerations about time and hardware required to run it. With such an algorithm, cancer specialists can widen their exploration studies on probable mRNA isoforms and their related proteins.

1.4 Thesis Overview

The thesis is structured to provide a good understanding of AS and its importance for cancer research, provide an overview of the state-of-the-art of Deep Learning models for genomics, explain the design choices of our algorithm, how we evaluate it, and our results. The following presents short details of what to find in each chapter.

- **Chapter 2: Background**
  It presents the bases of transcriptomics, from the sequencing to how the information encoded in genes becomes proteins. The mechanism of alternative splicing is explained and put into the context of cancer. And from the computational side, we present Deep Learning algorithms, Large Language Models, and six foundation models from the literature that have been pre-trained with either DNA or RNA data, so as to make a decision on which one to use in our task.

- **Chapter 3: Materials and Methods**
  Details on the IIGM's dataset and on the references used in the UCSC Genome Browser. Then, an explanation of how we performed the combinatorial tests, how we evaluated and processed data in an attempt to reproduce results from the paper of our chosen LLM model. Followed by the design of the first and second parts of our algorithm, and how our results are evaluated.

- **Chapter 4: Results**
  Information on the combinatorial tests on a couple of genes. Tests with the chosen LLM model to verify characteristics of the sequences generated purely by its Next Token Prediction (NTP), in terms of realistic bases and structures. The results of our algorithm first with the genes FOXA1, HRAS, MYC, KLF6, MDM2, SRSF1, then with 51 other genes. And the resources necessary to run it in terms of time and VRAM.

- **Chapter 5: Conclusion**
  It presents the key findings from our work and sketches future work on how to personalize our algorithm for each gene, to reduce some of its biases, and which directions to take to deepen the study of the algorithm and the produced isoforms.

Chapter 2

Background

This thesis combines knowledge from cancer transcriptomics and ML in order to create an algorithm able to create multiple mRNA isoforms from a gene, so from the biological point of view, we will introduce concepts of genomics, sequencing, alternative splicing, and how the latter relates to cancer. From the computational side, we discuss a brief history of Machine and Deep Learning models, then compare the state-of-the-art genomic models.

2.1 Transcriptomics

Transcriptomics is an interdisciplinary field that studies the complete set of RNA of an organism. It relates to genomics as the latter studies the function, structure, mapping, and editing of genomes, which are the sum of all DNA of a species, which differs from genetics, whose focus is on individual genes. In this section, we will understand how DNA and RNA are related and why it is important for understanding cancer and other diseases.

The monomeric unit of DNA and RNA nucleic acid polymers is the nucleotide, an organic molecule composed of phosphate, a pentose sugar, and a nitrogenous base, also known as nucleobase [7, 8]. DNA has nucleobases of four types: adenine (A), cytosine (C), guanine (G), and thymine (T); the latter is substituted for uracil (U) in RNA. The letter (N) is used to indicate any of the bases. For simplicity, only T will be used in this work. The Human Genome Project (HGP) estimates that our species has three billion base pairs [9].

2.1.1 Sequencing

A fundamental technology for genomics and transcriptomics is sequencing, the process of determining the order of nucleobases in a DNA or RNA molecule. The

two main sequencing technology classes are the short-reads, used for many years in the HGP, and the more recent long-reads. As the names suggest, the first is able to sequence shorter strands of DNA, i.e., a few hundred bases, and the latter longer strands, typically of $10 - 100$ knt [10]. There are multiple technologies available on the market for both classes, and we will present the most established short-read sequencing technology and the long-read one that was used to generate the IIGM dataset used in this thesis.

Short-read

Sanger Developed in 1977, Sanger sequenced the genome of bacteriophage ΦX174 using DNA polymerase under controlled conditions. The idea behind it is to insert a template sequence into four different capillary tubes, along with deoxynucleotide triphosphates (dNTPs) for all bases (dATP, dCTP, dGTP, dTTP), primers, and, for each container, different dideoxynucleotide triphosphates (ddNTPs) that terminate DNA synthesis [11].

In each tube, this process will synthesize sequences of different lengths, stopping at specific bases, e.g., tube 1 contains ddATPs and stops the synthesis only for the A base. Finally, electrophoresis is applied, and by comparing the four tubes, we observe a one-to-one relationship between position and base.

In 1986, Applied Biosystems commercialized a more efficient method using four fluorescent terminators as dyes, allowing a single tube to be sufficient for electrophoresis. This method enabled each position to be identified by color, facilitating the processing of results through a chromatogram [12].

Next Generation Sequencing (NGS) Also known as second-generation sequencing, NGS was another important step in sequencing, allowing the whole human genome to be sequenced in hours or a few days because of its massive parallelization of reads per run [13], compared to the couple of decades that were required by the previous method used in the HGP. Illumina is a major player in NGS, utilizing the Sequencing by Synthesis (SBS) method [14].

The method consists of fragmenting a large portion of DNA into small strands (length of a read), and each fragment is amplified in spatial clusters, representing the read. Then, sequencing begins with each read emitting a fluorescent light corresponding to a base; in clusters, these signals are clearer [15]. The limitation in the length of the reads is associated with chemical randomness of enzymes causing a phase error; the cluster signal may be noisier due to misalignment of strands within a cluster, e.g. in a cluster most strands are emitting the signal of the base at position n, while some strand might be emitting $n - 1$ or $n + 1$, the longer the strand, the more probable it is for this kind of drift to become an issue [16].

Long-read

Nanopores Developed by Oxford Nanopore Technologies (ONT), this technology uses an array of nanoscale protein pores (nanopores) within an electrically resistant polymer membrane, where a constant voltage is applied. A strand of DNA or RNA passes through the pore with the assistance of a motor protein, which controls the speed of its passage. When different bases cross the membrane, different currents are read. However, there are not only five signal values (A, C, G, T, none), but it is a more complex electrical signal because multiple bases are inside the nanopore at once, creating a signal processing challenge [17]. ONT provides a proprietary software solution called Dorado that relies on signal preprocessing, Machine Learning (ML) models, and postprocessing [18]. Accuracy can be improved by running multiple reads and creating a consensus sequence [17].

Comparison

Long-reads can optionally directly sequence RNA (dRNA), while the current short-read technology relies solely on the sequencing of complementary DNA (cDNA) instead of direct sequencing of the RNA, introducing biases related to the conversion process, which utilizes the reverse transcriptase enzyme and dNTPs [19]. Furthermore, the short-reads utilize the Polymerase Chain Reaction (PCR) to amplify copies of DNA transcripts, thereby improving the accuracy of the reads; this process also introduces PCR bias [20].

The greater lengths of long-reads bring an advantage when it comes to sequencing transcripts with structural mutations or repeated sequences.

One issue with Nanopore is that when sequencing the last ~ 15 nt, the motor does not control the strand anymore, and it passes through the pore rapidly, effectively losing the sequencing of such bases [21].

2.1.2 From genetic code to proteins

Our genome is composed of 23 chromosome pairs, each chromosome is a thread-like structure of chromatin, a mixture of proteins and DNA; one of these proteins is the histone, which provides structural support for the DNA to be wrapped around it. Closed chromatin (heterochromatin) is densely packed and does not allow for transcription. When the DNA is more loosely wrapped around the histones, we have an open chromatin (euchromatin), which provides for transcription. With an open chromatin, an enzyme called RNA polymerase can synthesize precursor messenger RNA (pre-mRNA) from a locus of the DNA sequence. The pre-mRNA is spliced into messenger RNA (mRNA), which leaves the nucleus of the cell into the cytoplasm, where a ribosome can translate the mRNA into a protein, if the mRNA is protein-coding [22]. Eukaryotes have specific RNA polymerase enzymes

Technology	Short-read sequencing	Long-read sequencing	
	Illumina	ONT-cDNA	ONT-dRNA
Accuracy	$\sim$ 99.9%	99.75% (v5 sup) 99.25% (v5 hac)	98.66% (v5 sup) 97.54 (v5 hac)
Cost	High	Medium	Medium
Time	Days to weeks	Real-time to days	Real-time to days
Strengths	High-throughput, high accuracy, well-established	Long-read, minimal assembly required	Long-read direct sequencing, No PCR bias
Weaknesses	Limited ability to capture long-read related information	Relatively lower accuracy than other cDNA-based methods, lower throughput	Higher error rate, lower throughput high RNA input required not well-established for short RNAs

Table 2.1: Comparison of sequencing technologies. (v5 sup) and (v5 hac) correspond to different base-calling technologies with the same chemistry kit. Adapted from Katapodi X. et al. 2025 [21] (CC-BY-NC 4.0)

for different types of genes, possibly producing also micro RNA (miRNA), small nuclear RNA (snRNA), small nucleolar RNA (snoRNA), transport RNA (tRNA) and ribosomal RNA (rRNA) [23]; but the focus of this thesis is on mRNA, and the general procedure described above is valid only for our purposes, the actual behaviour is much more complex.

2.1.3 Splicing

We will focus our energies on splicing, which takes as input a pre-mRNA and as output an mRNA. The pre-mRNA is a direct translation of the gene's DNA, while the mRNA cuts out pieces called introns and keeps fragments called exons [24]. In Fig. 2.1, we see the double-stranded DNA being transcribed into RNA, or pre-mRNA, followed by splicing; the figure already includes representations of different proteins resulting from alternative splicing.

Exons combined are the building blocks of the mRNA, so we can expect a certain structure for it to synthesize a protein in the ribosome. This translates triplets of nucleotides called codons into amino acids; namely, there are 64 (4^3) codons, 61 of which produce amino acids, from which Methionine (ATG) can also act as

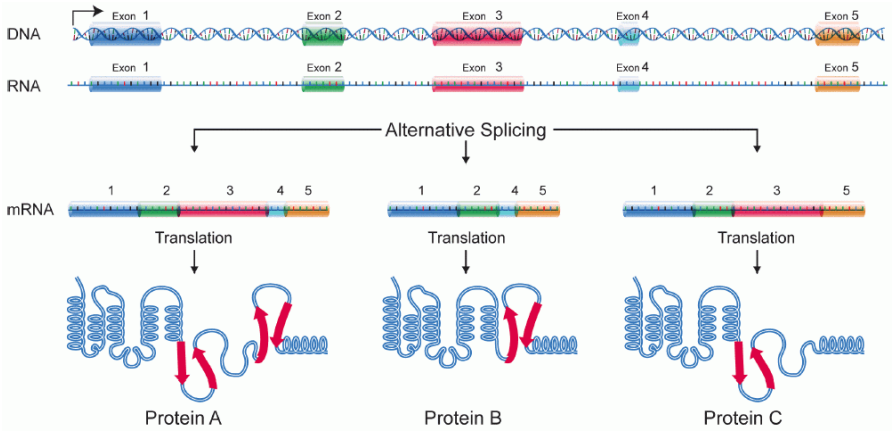

Figure 2.1: Alternative Splicing from DNA to mRNAs. Original from the National Human Genome Research Institute [25] (PDM 1.0)

the start codon. The remaining 3 are stop codons (TAA, TAG, and TGA), which terminate protein synthesis.

The Open Reading Frame (ORF) is the portion of DNA that begins with a start codon and ends with a stop codon (excluded). The number of bases inside the ORF must be multiples of three to respect the triplet constraint for each codon. Untranslated Regions (UTRs) are present at both ends of the mRNA; these non-coding sections exert regulatory functions, although their exact behavior is still under study [26].

Hence, the overall structure of a protein-coding mRNA, or the joined exons, must contain the 5′ UTR, a start codon, amino acid codons, a stop codon, and the 3′ UTR. The coding portion of exons tends to have a higher conservation across species, due to natural selection [27].

Introns, on the other hand, contain non-coding information, exert regulatory functions [28], and contain well-conserved splicing signals [29, 30].

Splicing mechanism

The spliceosome is a large ribonucleoprotein (RNP) complex that is found primarily within the nucleus of eukaryotic cells. Throughout this thesis, we will consider only the major spliceosome U2 and its splicing signals, since the minor spliceosome U12 accounts for less than 1% of all splicing in eukaryotes [31]. The small nuclear RNP (snRNP) that compose the major spliceosome are: U1, U2, U3, U4, U5, and U6; they are capable of binding to the pre-mRNA and catalyze specific reactions [32].

The U2 Splicing mechanism can be decomposed in several steps and complexes [32]:

- Splicing starts with the U1 binding to the 5' splice site (5'SS), composed of the sequence GT. Then, the branch point (BP), which is composed of an A, is recognized by and bound with a Branch Point Binding Protein (BBP), also known as Splicing Factor 1 (SF1). The 3' splice site (3'SS), which is composed of the sequence AG, is recognized by and bound with the U2 auxiliary factor (U2AF). An SR protein bridges between 5'SS and the BP. This ends the recognition phase by forming the Early complex (E-complex);

- Then, the U2 snRNP substitutes SF1, BBP, and U2AF, binding with the BP and the polypyrimidine tract (PPT) between BP and 3'SS, forming the pre-spliceosome complex (A-complex);

- When the tri-snRNP U4/U5/U6 joins this process, the U6 binds to U1 at the 5'SS, the U4 with the 3'SS, and U5 takes the role of bridging the exons, replacing the SR proteins, forming the pre-catalytic spliceosome complex (B1-complex);

- For it to be catalytically active, U1 and U4 leave, with U6 replacing U1 at the 5'ss, forming the catalytically active spliceosome complex (B2-complex);

- The first transesterification reaction occurs, where the A at the BP attacks to the G at the 3'SS, separating the 5' exon from the intron, the U5 keeps the exon in the complex, and the intron takes the shape of a lariat, overall forming the catalytic-1-complex (C1-complex);

- Followed by a second transesterification, where the 5' exon is ligated to the 3' exon, and cleaving the intron lariat. The exons form the mRNA, while the intron bonded to the remaining snRNPs is called the catalytic-2-complex (C2-complex).

2.1.4 Alternative Splicing (AS)

Alternative splicing is the process by which a single pre-mRNA can be spliced into multiple mRNA isoforms. We will briefly cover the main discoveries related to this phenomenon, its mechanisms, and how it relates to cancer and other diseases.

In 1941, Beadle and Tatum's "one gene–one enzyme hypothesis" was vital for understanding protein synthesis, winning the Nobel Prize in Physiology or Medicine in 1958 [33, 34].

In 1977, two articles presented the same conclusion that a single Adenovirus type 2 (Ad2) gene produced multiple mRNA, as observed through an electron microscope [35, 36].

Studies show that $90 - 95\%$ of human genes are naturally subject to AS, it is a natural phenomenon that increases the complexity of the proteome without

increasing the size of the genome [37, 38]. It has also been associated with cellular differentiation [39, 40]. However, dysregulation of AS can produce abnormal proteins and may cause diseases, such as neurodegenerative, autoimmune, and cancerous conditions, among others [41, 42, 43, 44].

Mechanisms of Alternative Splicing

AS is stochastic by design, depending on external proteins, external RNAs, and signals inside the sequence itself, i.e., 5'SS, BP, PPT, 3'SS, Exonic Splicing Enhancers (ESEs), Exonic Splicing Silencers (ESSs), Intronic Splicing Silencers (ISSs), and Intronic Splicing Enhancers (ISEs) [45].

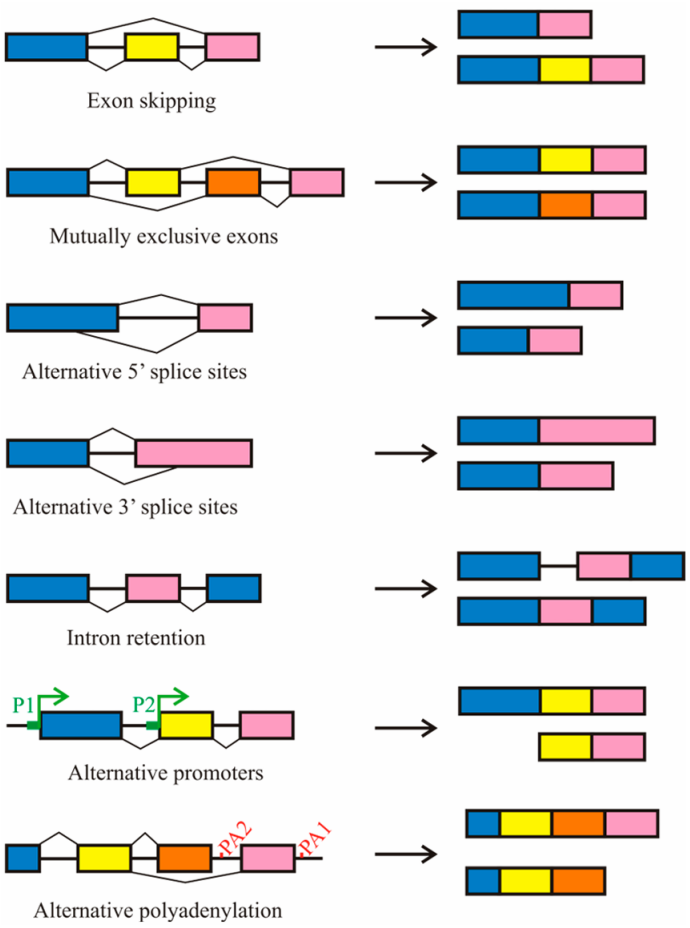

Figure 2.2: Mechanisms that produce different mRNA isoforms. Adapted from Gimeno-Valiente F. et al. 2024 [46] (CC BY 4.0)

There are seven known ways it occurs (Fig. 2.2); the latter two are not strict AS events, but are included because they produce different mRNA isoforms [47, 46]:

1. **Exon Skipping or Cassette Exon:** Exon e_j is skipped when the spliceosome splices between the 5' donor site of e_{j-1} and the 3' acceptor site of e_{j+1}, splicing out together the introns i_{j-1} and i_j;
2. **Mutually exclusive exons:** Only one cassette exon is retained among an array of two or more exons;
3. **Alternative 5' donor site:** It splices at a different 5'SS, changing the 3' boundary of the upstream exon;
4. **Alternative 3' acceptor site:** It splices at a different 3'SS, changing the 5' boundary of the downstream exon;
5. **Intron Retention:** An intron is not spliced out;
6. **Alternate promoters:** Alternative starting point for transcription;
7. **Alternate polyadenylation:** Alternative ending point for transcription.

2.1.5 Alternative Splicing in Cancer

The dysregulation of AS may induce cancer [43, 48, 49, 4, 50, 2]. Mutations that cause changes in AS inside the same gene are called cis-mutations. These can be structural, i.e., deletions, duplications, inversions, insertions, and translocations. Additionally, there are single-nucleotide mutations, i.e., substitutions, insertions, and deletions. Some examples of point cis-mutations are:

- Loss of 3'SS: AG $\rightarrow$ TG;
- Protein codon is mutated into a stop codon: TCA $\rightarrow$ TAA;
- Insertion or deletion inside exons of the ORF may create non-coding isoforms, due to the shift of the ORF.

In cancer, trans-elements play an important role in the dysregulation of AS too [4, 50]. SF3B1 is the most frequently mutated spliceosomal component in cancer. It is a component of the U2 snRNP; its function is to recognize the BP and to assemble the A-complex. Such mutations lead to misrecognition of the BP, consequently of the 3'SS, different cassette exon inclusion, and reduced intron retention. Another example is SRSF2, an SR protein that mediates exon inclusion and recognition of 5'SS and 3'SS by interacting with U1 and U2 during splicing; one of its mutations recognizes C-rich sequences, but has reduced affinity for G-rich sequences, while a wild SRSF2 recognizes both, resulting in mis-splicing [4].

Both cis-mutations and trans-elements can upregulate oncogenic isoforms or downregulate tumor suppressor isoforms. AS has various roles in cancer stages [2], for example:

- Upregulation of ITGA6, PKM2, NUMB-PRR(L) and downregulation of NUMB-PRR(S) contribute to the proliferation;
- Upregulation of Bcl-x(s) and downregulation of Bcl-x(l), suppress apoptosis;
- Upregulation of PKM2 and downregulation of PKM1 alter the metabolism of the cell;
- Upregulation of VEGF-A165b takes part in angiogenesis;
- Upregulation of TAK1 and CD44(s) contribute invasion and metastasis.

Nonsense-Mediated Decay (NMD) is a surveillance mechanism that degrades abnormal mRNA, but it can be evaded by AS in some ways, e.g., altering splice sites, modulating mRNA stability, changing mRNA secondary structure, and changing the interaction with RNA-binding proteins (RBPs). In addition, some cellular stresses may inhibit NMD action, such as, amino acid deprivation, hypoxia, nutrient deprivation, infection, reactive oxygen species, and double-stranded RNA. Tumours dynamically regulate NMD to adapt to the tumor microenvironment (TME), where hypoxia and nutrient deprivation prevail, promoting survivability, proliferation, and metastasis. NMD may even promote cancer progression, as in myelodysplastic syndromes (MDS), a mutation on SRSF2 leads to isoforms containing premature termination codons (PMCs), leading to the downregulation of EZH2 and INTS3 genes; such depletion synergizes with the RAS pathway, leading to malignant proliferation and to the transformation from MDS to acute leukemia [3].

2.1.6 Clinical implications

Knowing which alternatively spliced isoforms are produced in cancer cells is important in cancer treatments, not only for understanding the disease, but also directly usable in therapies [3].

In breast and colon cancer, there is the overexpression of serine-arginine protein kinase 1 (SRPK1), a splice regulatory protein [51]. *In vitro*, in cells treated with chemotherapy drugs, the downregulation of SRPK1 by transfection of sh1-SRPK1, a siRNA (small interfering RNA), increased apoptosis of the carcinomas, compared to the transfection of a control plasmid. The downregulation of SRPK1 changes the splicing of MAP2K2. The new isoforms were by either coding deletion (Δexons$7-8$) or loss-of-frame (Δexons7 and Δexons$7-10$), isoforms whose functions are to be determined. MAP2K2 wild isoforms encode the two major kinases responsible for the phosphorylation of MAPK3 and MAPK1 proteins. These genes are on the protein pathways necessary for cell proliferation and resistance against chemotherapy drugs [52].

Some AS events unique to cancer can create proteins with new epitopes, which act as neoantigens that can be exploited in immunotherapies, especially in cancers with low tumor mutational burden (TMB). Using genetically modified chimeric antigen receptor T cells (CAR-Ts) to recognize and attack neoantigens is a promising cancer therapy [3, 53, 54].

2.2 Deep Learning and Large Language Models

In order to create mRNAs from DNA gene sequences, we will use statistical learning enabled by algorithms of Deep Learning (DL), so in this section, we will introduce some of its main models and concepts.

Machine Learning (ML) is a field of study in Artificial Intelligence (AI), with statistical algorithms for predicting discrete or continuous values that can learn from a *training dataset*, and then generalize its results to unseen data. DL is a subset of ML algorithms that rely on an architecture inspired by our brain, Artificial Neural Networks (ANN) with multiple hidden layers, which usually provide better metrics in more complex tasks [55]. In recent history, some models have revolutionized the state-of-the-art when solving particular tasks, so we present some of them and how they were used.

Convolutional Neural Network (CNN)

One of the first ANNs with special operators in the neurons that gained much attention was the CNN. Capable of reading hand-written English characters, LeNet-5 [56], was used for years to automatically read cheques. Neurons perform multidimensional convolution operations, instead of simple matrix multiplications. This change allowed for spatial dependencies to be learned, and the number of learnable weights per layer was reduced, meaning more layers could be introduced, delivering more abstraction for less computational effort.

Recurrent Neural Network (RNN)

This architecture is specifically designed to handle sequential data, such as natural language, RNA, and stock prices, through the use of a summing ReLU-activated feedback loop that adds up information from the past to make predictions of the following data points. One issue with this design is that for long sequences, the feedback loop can cause either an exploding or a vanishing gradient problem [57], depending on the weight of the loop, $w > 1$, and $w < 1$, respectively.

Long Short-Term Memory (LSTM) networks [58] were developed to address the vanishing/exploding gradient problem and to better utilize input data by separating

the paths of closest context from the furthest, utilizing different activation functions such as hyperbolic tangent and sigmoids.

Large Language Models (LLM)

An improvement for models dealing with sequential data, the name comes from the number of parameters, reaching hundreds of billions, and the task domain it was originally used for, Natural Language Processing (NLP) [59]. It substituted RNNs in NLP for two main reasons: the ability to create input-dependent embedding spaces and the parallelization of computations. In the following paragraphs, we will present the two operators used by the model we chose to incorporate into our algorithm: Attention, the operator that has brought impressive improvements to NLP in recent years, and Hyena Hierarchy, a new operator that surrogates Attention with a reduced computational load.

Attention Transformers [60] are models that are based solely on the Multi-Head Attention mechanism through an encoder-decoder structure, with global context unless restricted. Self-Attention takes as input the query Q, key K, and value V, which are projections of the input data at the first layer (Eq. 2.1).

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

d_k is the projection dimension for the key and query, which is used in the denominator as a scaling factor to avoid vanishing gradients of the softmax function. d_v is the output value dimension. Multi-Head Attention puts h heads in parallel (Eq. 2.2).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2.2}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.3}$$

We have learnable parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$,$W_i^K \in \mathbb{R}^{d_{model} \times d_k}$,$W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, with d_{model} the embedding dimension of the inputs. Each Self-Attention has Complexity $O(L^2 \cdot d_{model})$ with L the sequence length.

To make use of order and position, the model requires a positional encoder. The original Transformers paper proposes a new embedding of sine and cosine functions (Eq. 2.4).

$$\begin{cases} PE_{(pos,2i)} & = sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} & = cos(pos/10000^{2i/d_{model}}) \end{cases} \tag{2.4}$$

Hyena Hierarchy The Hyena operator is introduced in 2023 [61], matching metrics of attention-based models, with sub-quadratic scaling on sequence length, enabling longer context information. It is based on interleaving implicitly parametrized long convolutions and data-controlled gating. Its training time is 100× faster than the highly optimized FlashAttention [62] at sequence length 64K.

A discrete convolution is defined in Eq. 2.5.

$$y_t = (h * u)_t = \sum_{n=0}^{L-1} h_{t-n} u_n \tag{2.5}$$

With input signal u of length L, and filter h which is measurable in $L^1(\mathbb{Z})$ sense $\sum_{t=-\infty}^{\infty} |h_t| < \infty$, with learnable parameters of the model. This computation can be substituted by the Toeplitz kernel matrix $S_h \in \mathbb{R}^{L \times L}$ (Eq. 2.6).

$$(h * u) = \begin{bmatrix} h_0 & h_{-1} & \dots & h_{-L+1} \\ h_1 & h_0 & \dots & h_{-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{L-1} & h_{L-2} & \dots & h_0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{L-1} \end{bmatrix} = S_h u \tag{2.6}$$

However, the computational cost of normal convolutions is $O(L^2)$, therefore, Fast Fourier Transform (FFT) [63] is used to reach $O(L \log_2 L)$ asymptotic cost without materializing S_h. In addition, h is implicitly defined by a family of parametrized functions $h(t) = \gamma_\theta(t)$, where θ are the parameters of the function γ_θ. The class of functions is a design choice, instead of defining $h(t)$ as a Finite Impulse Response (FIR), whose number of parameters would scale linearly with the memory range of h. A couple of implicit parametrization examples are the family of state-space models (SSM) and feed-forward networks. The number of parameters is disentangled from the memory extent; rather, it defines the expressivity of γ_θ.

Hyena is defined as a class of data-controlled operators consisting of a recurrence of multiplicative gating interactions and implicit long convolutions. Let $(v_t, x_t^1, ..., x_t^N)$ be projections of the input and $h_0, ..., h_N$ a set of learnable filters, the Hyena of order N operator is defined by the recurrence:

$$z_t^1 = v_t \tag{2.7}$$
$$z_t^{n+1} = x_t^n (h^n * z^n)_t : n = 1, ..., N \tag{2.8}$$
$$y_t = z_t^{N+1} \tag{2.9}$$

Alternating products and convolutions with projections of the input data might be the strength of this operator, for the convolution in the time domain allows for a broader context, and element-wise multiplications in the time domain for a more fine-grained selection of components, as we can observe in Eq. 2.10.

$$y_t = x_t^N(h^N * (x_t^{N-1}(h^{N-1} * (\dots))))$$ (2.10)

The original Hyena hierarchy model is based on the H3 mechanism [64], which creates a surrogate for attention, while Hyena further generalizes to more projections of the input data.

$$A(q,k) = D_q S_\psi D_k S_\phi$$ (2.11)
$$H3(q,k,v) = A(q,k)v$$ (2.12)

With the Toeplitz matrices $S_\psi S_\phi$ of learnable causal filters parametrized by SSMs. The data projections are not limited to 3, indeed, we can extend it to N projections, all used in the Hyena recursion. Let $D_x^n = diag(x^n) \in \mathbb{R}^{L\times L}$, and S_h^n be the Toeplitz matrices corresponding to h_n, we get

$$y = H(u)v = D_x^N S_h^N D_x^{N-1} S_h^{N-1} \dots D_x^1 S_h^1 v$$ (2.13)

2.3 Deep Learning in Genomics

We shall present the state-of-the-art of genomic and transcriptomic foundation models, which can be adapted to perform a wide variety of tasks. They contain different operators, tokenization strategies, and training datasets. Afterwards, we discuss our choice of the foundation model to be used in our case.

2.3.1 Bidirectional Encoder Representations from Transformers model for DNA (DNABERT)

DNABERT [65] uses the Transformer architecture to get longer contexts with respect to RNNs and CNNs, although restricted to avoid a complexity explosion. The length of the DNA code was sampled at variable lengths between 5 and 510 bases, which does not take full advantage of our long-reads.

Tokenization

Instead of single-nucleotide resolution, they have opted for tokenizing in k-mer representation, creating groups of length k nucleotides with overlap, e.g., ATGTTC tokenized in 3-mers: {ATG, TGT, GTT, TTC}. Varying k, they have created pre-trained models with k set to 3, 4, 5, and 6, respectively, DNABERT-3, DNABERT-4, DNABERT-5, DNABERT-6. For specific tasks, fine-tuning was followed.

2.3.2 HyenaDNA

HyenaDNA [66] is a foundation model pre-trained on the human reference genome using next token prediction (NTP), and a context size of 1M bases. Its ability to predict splicing sites is measured in a classification task of sequences 400 bases long, with a possible site in the middle, after fine-tuning. It achieves comparable metrics with the Nucleotide Transformer Benchmarks, with F1-score of 96.6, 97.3, 97.9 respectively on the benchmarks for Splice acceptor, donor, and both. The model has 1.6M parameters, and was pre-trained on 1 genome. It is based on the H3 architecture [64].

Tokenization

HyenaDNA tokenizes at single-nucleotide precision at a DNA level; the tokens are {A, C, G, T, N}, the latter being any nucleotide.

2.3.3 BigRNA

BigRNA [67] is a commercial foundation model whose input is a DNA or RNA sequence, and it outputs a tissue-specific simulation of an RNA-seq experiment, i.e., the expression level of the mRNA. The model also performs well on the prediction of pathological mutations that lead to intron retention, exon skipping, and polyA site shift.

The overall architecture is an improvement to the Enformer [68], a model made out of a Convolutional layer, followed by a Multi-Head Attention, for it is an ensemble of 7 models trained with different hyperparameters and it has a resolution of 128 base pairs, a receptive field of 192 knt, and a total of 1.8 B parameters.

Tokenization

The tokens used by the MHA are not directly related to the sequence's nucleotides, but rather to abstract vector spaces output by the convolution and pooling of both sequence and RNA-seq.

2.3.4 SpliceBERT

SpliceBERT [69] is based on bidirectional encoder representations from transformers (BERT), self-supervisely pretrained using mask language modeling (MLM) on sequences of varying length between 64-1024. The paper demonstrates the importance of cross-species training for the aggregation of evolutionary information about splicing sites. It was fine-tuned on the Spliceator dataset [70], trained on

the classification of sequences 400 nucleotides long, with a possible splicing site in the middle.

Tokenization

Single-nucleotide resolution, {A, C, G, T, N}, with [CLS] and [SEP] added to the edges of sequences, as it is routine for BERT-style tokenizers.

2.3.5 Long read RNA with Striped Hyena (LoRNA$^{\mathbf{SH}}$)

LoRNA$^{\mathrm{SH}}$ [71] is a NTP model, with the StripedHyena architecture, a combination of the Hyena operator [61] and rotary self-attention, which is advantageous with respect to others foundation models for its complexity scalability $O(NDL(\log_2 L + D))$, with N the order of the Hyena operator, D the model width, and L the sequence's length. Models solely reliant on self-attention, such as the Transformers, although being powerful for computing pair-wise interactions, have a computational cost of $O(L^2)$, limiting the amount of context for the model. LoRNA$^{\mathrm{SH}}$ uses 65K nucleotides of context; it was pre-trained on 300K Human and Mouse long-read transcripts, totalling more than 7B tokens.

Tokenization

The 16 tokens utilized by LoRNA$^{\mathrm{SH}}$ give direct advantages for our task of predicting isoforms, for it has tokens distinguishing introns from the exons: $\{a, c, u, g\}$] for introns, $\{A, C, U, G\}$ for exons, $\{W, X, Y, Z\}$ represent non-RNA (DNA) genetic code flanking the transcripts, S the transcript start site (TSS), E the polyA site (PAS). Two other tokens specify the species of the sequence, H for human, and M for mouse.

StripedHyena Architecture

LoRNA$^{\mathrm{SH}}$ leverages the StripedHyena architecture for long-sequence language modelling, which integrates Hyena operators with rotary self-attention mechanisms. This architecture is designed as a series of transformations applied to an input sequence, denoted as $X \in \mathbb{R}^{L \times D}$, where L represents the sequence length and D is the dimensionality of the model's hidden states. Each layer in StripedHyena alternates between Hyena layers and rotary self-attention layers, combining the advantages of convolutional and attention-based approaches.

The LoRNA$^{\mathrm{SH}}$ model comprises 16 blocks, each with a model width of 128 dimensions. These blocks feature sequence mixing and channel mixing layers, allowing the model to handle information along both the sequence and model width dimensions. Specifically, the sequence mixing layers consist of 13 Hyena layers

interspersed with 3 evenly distributed rotary self-attention layers. The channel mixing layers utilize gated linear units to enhance performance. Furthermore, root mean square layer normalization is applied to the inputs of each layer for improved stability and consistency. There are a total of 3.3 million parameters, and a memory allocation for the model of around 6 MB.

2.3.6 Evo 2

Evo 2 [72] creates the multi-hybrid architecture StripedHyena 2, a specialization of the previous version, which introduces Short Explicit(SE), Middle Regularized (MR), and Long Implicit (LI) Hyena operators. The striped pattern is Hyena-SE, Hyena-MR, Hyena-LI, and Attention. The context range for the Hyena operators are, respectively, 7, 128, and either 8192 or 1 million base pairs, depending on the version of the released model, either with 7B or 40B parameters.

The model was pre-trained on OpenGenome2 [73], a 9.3T single-nucleotide resolution tokens from genomes of multiple species from prokaryotic, eukaryotic, non-redundant metagenomic sequencing data, and organelles. However, only 2T tokens were used to train Evo 2 7B.

They introduce an exon/intron classifier with the extraction of embeddings and train binary classifiers for each layer; the best layer on validation is retrained. Evo 2 7B is used, with Area Under the Receiver Operating Characteristic curve (AUROC) 0.82 for *Homo Sapiens*, which was held out from this training.

Tokenization

Evo 2 tokenizes at single-nucleotide precision at a DNA level; the tokens are {A, C, G, T, #, and @ }, the last two are special tokens that join indefinitely far sequences, and join contig sequences from the same strand that are near each other, respectively.

2.3.7 Our use case

In order to produce mRNA isoforms from a DNA sequence, the proposed algorithm relies on the probability of the NTP to make a localized decision of whether the following bases are introns or exons. These many generated isoforms are combined to create a normalized exon expression for each nucleotide in the sequence, and a simple binary classifier based on thresholds of such expression is used. So we require a model with specific characteristics, such as:

- Single-nucleotide resolution, because it would not make sense to classify k-mers as introns or exons, since this lower resolution would inevitably classify splicing signals as exons, despite them always being part of the intron.

- Long context, with thousands of bases, due to the three-dimensional nature of the splicing machinery, strand loops allow for the interaction with distal ESE, ESS, ISS, and ISEs [74].
- Straightforward probabilities related to exon or intron tokens to take full advantage of the complex architecture of foundation models, instead of relying on feature extraction to train simpler ML/DL models that classify between these classes, as Evo2 did.
- Trained on mRNA data, to avoid biological bias related to other sequencing methods, and preferably cancer, for it is the disease we are interested in.

LoRNA^{SH} matches all of these requirements; it has a single-nucleotide resolution; its context reaches up to 65 knt; the tokens are already trained to distinguish between introns and exons; and it was pre-trained on long-read mRNA from 26 cell lines of 9 different cancer tissues. While other models fail to fulfill at least one requirement.

We found only Evo2 to solve our task of exon classifier, and as described, they use a simpler model with the features extracted from their foundation model; using our algorithm with a model that tokenizes both introns and exons has overcome their AUC of 0.82 in our smaller test set. Other models usually solve a similar task, but with an important difference: they predict the probability of a position being a splicing site, which would induce the presence of an exon upstream or downstream; however, they do not take into account which bases are already classified as exon, they act on the DNA or RNA level, not on mRNA as LoRNA^{SH} does.

Chapter 3

Materials and Methods

3.1 Datasets

3.1.1 IIGM's Dataset

The Italian Institute for Genomic Medicine shared with us a dataset that is yet to be published; it contains long-read direct-mRNA sequencing using the GridION sequencer, which utilizes the Nanopore technology [75]. Seven replicates of PC3 were sequenced under heterogeneous conditions. PC3 is a commercial cell line of bone metastasis of a grade IV prostatic adenocarcinoma from a 62-year-old white male [76]. Expression is normalized by Transcripts per Million (TPM) [77], and for the rest of this thesis, we consider the long-read expression as a single number, being the average expression level across replicates. Known and novel isoforms are defined relative to the reference genome GRCh37 [5].

The dataset contains 11.111 different genes, 31140 isoforms from which 15819 (50.80%) are *known*, 14777 (47.45%) are *novel*, and 544 (1.75%) are classified as *others*.

The distribution of unique genes in the dataset is presented, by length, in Fig. 3.1, and by chromosome, in Fig. 3.2.

3.1.2 Genome Browser

The UCSC Genome browser was used with the GRCh37/hg19 reference [5] for two different tasks: aligning pure LoRNA^{SH} generations with BLAT [78] and visually comparing our algorithm's isoforms with known ones [79].

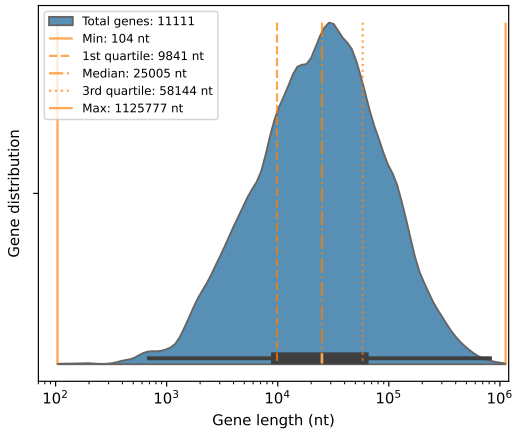

Figure 3.1: Gene length distribution on IIGM's dataset

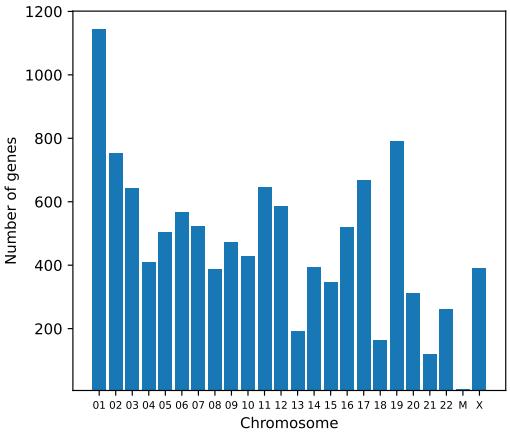

Figure 3.2: Chromosome distribution on IIGM's dataset

BLAT

BLAT stands for "BLAST-like alignment tool"; Basic Local Alignment Search Tool (BLAST) [80], and BLAT both scan short matches (hits), extend these in high-scoring pairs (HSP), and create indexes of sequences. However, BLAT is faster than BLAST because the former creates k-mer indexes of the whole genome and scans linearly on the query sequence, while the latter creates k-mer indexes of the query sequence and scans linearly over the genome. In addition to other improvements, an important one for our study is that BLAT performs better on aligning exons, without creating wrong intersections with introns.

Since indexing the genome is slower, the tool is available online on the UCSC Genome Browser; for code automation, we used its REST API. The pure LoRNA^{SH} generations were aligned in two ways to the reference, either considering the whole sequence (i.e., RNA), or only the exons (i.e., mRNA), and only the maximum match is considered in our analysis. The minimum number of matched bases in BLAT is 20 nt.

Genome Browser tracks

GENCODE V48lift37 track is a high-quality and manually curated dataset of whole human genome annotations, generated by GENCODE [81]. It is a merger between the manual Human and Vertebrate Analysis and Annotation (HAVANA) [82] and the automatic annotation pipeline of Ensembl [83].

phyloP is part of the Phylogenetic Analysis with Space/Time models (PHAST) project [84]; this track outputs the P-value, which means *conservation* when positive, so it is highly conserved across 100 vertebrates or *acceleration* when negative, so it is expected to chang, given the other species' genomes. It is based on a Phylogenetic Hidden Markov Chain (phylo-HMM) [6].

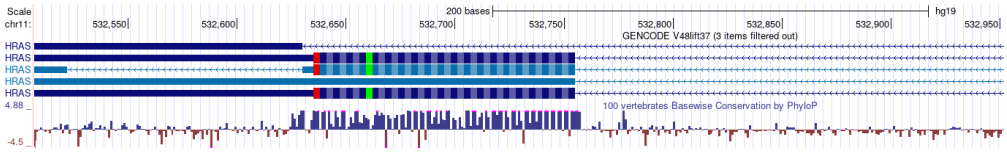

Figure 3.3: Genome Browser example. On the first row, there are the scale (200 bases) and the genome reference (hg19). On the second row, the coordinates with chromosome (chr11) and position (e.g. 532,550). From the third to the eighth row we observe the GENCODE V48lift37 track: We see the gene name (HRAS) and five isoforms; the solid lines represent exons and the most narrow lines, introns; the arrows on the introns is the strand (-); the narrower solid lines represent untranslated regions of an exon; the stripes on the coding exons are each codon; the green and the red codon are, respectively, the start and stop codons. Below we have the 100 vertebrate Basewise Conservation by PhyloP track; We observe higher conservation of the coding exons and the splicing site.

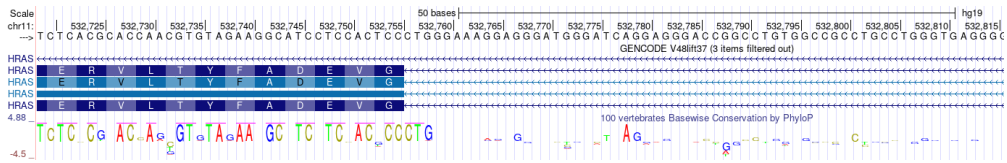

Figure 3.4: Genome Browser example - Zoom. With a smaller scale (e.g. 50 bases), we can see the nucleotides of the reference just below the coordinates. On the GENCODE V48lift37 track, we observe the corresponding amino acid of each codon. On the PhyloP track, we notice a higher conservation at most bases in the coding exons and at the CT 3'SS (AG on the positive strand).

3.2 Generation of isoform with splicing signals

These methods aimed to generate all possible isoforms s^i given a pre-mRNA sequence S. All make use of recursive algorithms and start as exons, and all work with this change of state between introns and exons; the difference among them is when to consider making a process split. The output encodes bases in introns and exons as lower and upper letters, respectively, as in 2.3.5.

Many sequences s are created, plus one at every split. Namely, if a sequence s^i splits, the created sequence will be s^j, with $j > i$. Each sequence is generated through a different process; these were computed in parallel, with a variable maximum number of parallel processes, defaulting to 48.

The computational costs follow the same linear-exponential law, but with different exponents. L is the length of S, and $c = c(S)$ is the variable coefficient among the different algorithms; attempts were made to reduce c, trading off with the probability of generating real isoforms in a reasonable time.

$$\mathcal{O}\left(L \cdot 2^c\right) \tag{3.1}$$

3.2.1 Basic splicing signal motifs

Firstly, we consider the three main splicing signals, the 5'SS (GT), the BP (A), and the 3'SS (AG). Beginning s_0^i as exon, we iterate a window of size 2 nt, with stepsize of 1 nt, if a the donor basic motif GT is encountered at some position p, a split is made, and another process is created with a new sequence s^j where s_p^j is an intron, while s_p^i remains as exon. Symmetrically, when in intron state, first we find a BP, then when an AG is found at position p^*, we split the process again, with s^j continuing the sequence as intron, and the new sequence $s^k : k > j$ is so that the sequence continues as an exon, meaning that in s^k, it splices out a possible intron from positions p to p^*.

Therefore, $c(S)$ is upper bounded by the number of GT (d) and AG (a) subsequences in S.

$$\Theta\left(L \cdot 2^{d+a}\right) \tag{3.2}$$

Most probable motifs

To reduce c, a reasonable approach is to expand the window size of the splicing signal motif check, and split only if such a motif is likely observed in nature.

The conservation of splice signal motifs was obtained from the work of Guigó Lab [31], with Position Weight Matrices (PWMs). Each position has a probability score for each base, with respect to the SS of orthologous U2 introns in human, mouse, rat, and chicken. Values for 5'SS are available on Tab. 3.1 and 3'SS values on Tab. 3.2.

We have changed the window size of the 5'SS motif to 5 nt, from positions -1 to 4. The window size of the 3'SS motif went to 4 nt, from positions -3 to 1. Such decisions were made by the probability of the bases in these positions, all are above 60% or 35% for donor and acceptor, respectively. The length of the window size is still small, to avoid creating motifs with probabilities that are too similar, and to limit the number of different motifs.

#	A	C	G	T
-3	0.341	**0.359**	0.181	0.119
-2	**0.638**	0.11	0.111	0.141
-1	0.1	0.029	**0.802**	0.069
0	0	0	**1**	0
1	0	0	0	**1**
2	**0.609**	0.026	0.339	0.027
3	**0.707**	0.074	0.111	0.108
4	0.082	0.053	**0.794**	0.072
5	0.174	0.15	0.19	**0.487**
6	0.3	0.196	**0.293**	0.211
7	0.223	0.252	0.238	**0.287**
8	0.218	0.268	0.243	**0.271**
9	0.222	0.245	0.255	**0.278**
10	0.221	0.25	0.259	**0.271**
11	0.214	0.243	0.256	**0.287**
12	0.207	0.251	0.252	**0.289**
13	0.215	0.246	0.25	**0.289**
14	0.214	0.245	0.253	**0.288**
15	0.215	0.235	0.26	**0.29**
16	0.213	0.237	0.264	**0.286**
17	0.216	0.237	0.26	**0.288**
18	0.214	0.24	0.256	**0.29**
19	0.217	0.239	0.257	**0.287**

Table 3.1: Donor PWM. Negative positions are exons, non-negative are introns.

The probability of a motif is calculated using the product rule, assuming independence, as in the study. Considering a motif subsequence S.

$$P(S) = \prod_{i}^{L} \mathrm{PWM}_i(S_i) \tag{3.3}$$

The number of non-zero probability motifs for donor and acceptor sites was, respectively, 256 and 64. The top 10 motifs are displayed in Tables 3.3 and 3.4, while the distribution of probabilities is displayed in Fig. 3.5. We notice that using this method, there is a huge majority of motifs whose probability is below 0.1%, but as we will see in the results, true isoforms may contain motifs that are very low in this probability ranking.

For both splice sites, we ordered the motif probabilities, and when iterating through the nucleotides in S, we check either the probability of the motif and

#	A	C	G	T
-20	0.195	0.263	0.153	**0.388**
-19	0.179	0.267	0.152	**0.402**
-18	0.158	0.275	0.148	**0.42**
-17	0.141	0.277	0.144	**0.438**
-16	0.13	0.28	0.138	**0.452**
-15	0.117	0.289	0.132	**0.462**
-14	0.106	0.286	0.126	**0.482**
-13	0.097	0.282	0.12	**0.502**
-12	0.088	0.282	0.109	**0.521**
-11	0.079	0.265	0.104	**0.552**
-10	0.08	0.285	0.11	**0.525**
-9	0.091	0.3	0.115	**0.495**
-8	0.102	0.334	0.105	**0.458**
-7	0.106	0.344	0.09	**0.46**
-6	0.081	0.353	0.061	**0.505**
-5	0.083	0.301	0.06	**0.556**
-4	0.244	0.271	0.208	**0.277**
-3	0.054	**0.655**	0.002	0.289
-2	**1**	0	0	0
-1	0	0	**1**	0
0	0.253	0.141	**0.493**	0.113
1	0.241	0.189	0.199	**0.371**
2	0.259	0.236	0.236	**0.27**

Table 3.2: Acceptor PWM. Negative positions are introns, non-negative are exons.

if above a certain threshold P_t, we split. Another method was also applied; we consider the motif's rank compared to the others, and s^i would split with a motif above some threshold rank R_t.

Regular expression on motifs

A way of exploring the bases around the basic mononucleotide of the BP is to use a regular expression provided by Xie J. et al. 2023 [85] for humans. It was observed that compared to yeast, which has a relatively fixed BP motif TACTA<u>A</u>C, humans have evolved to accept a variety of motifs, providing more AS capabilities. The proposed motif is YTN<u>A</u>Y, in IUPAC code, Y means pyrimidine (C or T) and N means any base [86].

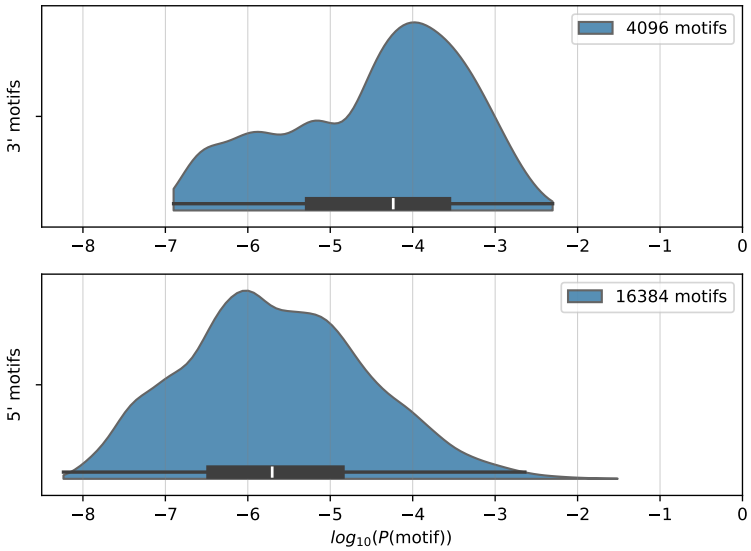

Figure 3.5: Distribution of 3'SS and 5'SS motif probabilities

Top	Motif	Probability
1	cag**GT**AAGT	3.058%
2	aag**GT**AAGT	2.905%
3	cag**GT**GAGT	1.702%
4	aag**GT**GAGT	1.617%
5	gag**GT**AAGT	1.542%
6	cag**GT**AAGG	1.193%
7	aag**GT**AAGG	1.133%
8	cag**GT**AAGA	1.093%
9	aag**GT**AAGA	1.038%
10	tag**GT**AAGT	1.014%

Table 3.3: Top 10 Donor motifs. In lower case exons, in upper case introns, in bold the splicing site

Distance between Branch Point and Acceptor

Considering the physical dimension of the SF1 and U2AF, which bind, respectively, to the BP and 3'SS, the distance observed between them in spliced lariats in humans follows the distribution presented in Fig. 3.6. The expected interval for constitutive splicing is 10-60 nt, 9% were found below 10 nt (proximal), and 8% above 60 nt (distal). In the figure, it is also illustrated how distal BP may produce

Top	Motif	Probability
1	TTC**AG**gtt	0.498%
2	TTC**AG**gtt	0.487%
3	TTC**AG**gta	0.478%
4	TCC**AG**gta	0.468%
5	TAC**AG**gtt	0.439%
6	TTC**AG**gtc	0.435%
7	TTC**AG**gtg	0.435%
8	TCC**AG**gtc	0.426%
9	TCC**AG**gtg	0.426%
10	TAC**AG**gta	0.421%

Table 3.4: Top 10 Acceptor motifs. In lower case exons, in upper case introns, in bold the splicing site

exon skipping and proximal BP, alternative 3'SS.

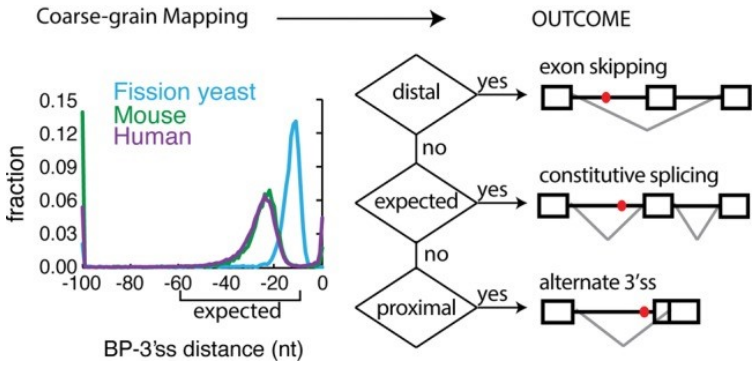

Figure 3.6: Distance distribution between branch and acceptor splice signals. Adapted from Taggart J. et al 2017 [87] (CC BY-NC 4.0)

3.3 LoRNA^{SH} Pure Generations

We put to the test LoRNA^{SH}'s ability to generate realistic isoforms through NTP, with initial prompt $x = \text{HS}$, and the maximum length of each generation is set to 20.000 nt, as in the original paper [71]. We call them *pure generations*, because there is no interference from our algorithm. And we studied the computational limits of LoRNA^{SH} on some GPUs.

3.3.1 Introns and Exons Statistics

We take all pure generations and filter out sequences smaller than 30 nt (small), that had not reached the end token until the position 20 knt (big), with continuous DNA regions at the end (big DNA), or that contain less than 30 exon tokens (full intron).

We count the number and lengths of introns and exons for each pure generation. We define the values for minimum intron and exon length to consider anything less than that as noise. Every noise, be it intron, exon, or DNA, is skipped and does not interfere with the count and length of the interrupted region. These values were not based on the evidential minimum lengths of these classes, so to avoid adding bias to our evaluation of realistic structures present in the pure generations, rather, they were just made to filter out noise, so smaller numbers are chosen: $\min_{exon} = 3$ nt, the size of a codon, and $\min_{intron} = 5$ nt, the minimum bases needed for the 5'SS, the BP and the 3'SS (GTAAG). Introns that are at the end or beginning of the sequences are ignored, for they do not have any biological meaning, and their lengths could interfere with the length statistics. Besides that, they create counters that are unbiological, since $n_{exons} = n_{introns} + 1$.

The length of the whole sequence is also provided, filtering out the noise and the flanking introns as specified in the previous paragraph, and we present them in two ways: the basic sequence length and the mRNA length, the latter including only the exons.

When plotting the dataset's statistics, we needed an alignment of the mRNA sequences with the gene sequence to separate the introns from the exons, since only the exons are sequenced. Using GRCh37/hg19 as a reference to the alignment, we were able to align 17.863 (57%) using the `Biopython` library [88], which implements several alignment algorithms such as Needleman-Wunsch, Smith-Waterman, Gotoh (three-state), and Waterman-Smith-Beyer global and local pairwise alignment algorithms, and the Fast Optimal Global Alignment Algorithm (FOGSAA) [89, 90, 91, 92, 93]. We used the parameters: `mode = global`, for a global optimization, `match score = 1`, defining a unit, `mismatch score = -100`, avoiding single base mutations, `gap score = 0`, the gap represents the intron, `open gap score = -14`, to open introns only if gain an alignment of 14 points doing so, and `extend gap score = 0`, because extending the intron should not be penalized. After obtaining the alignments, we iterate over the possible alignments with the top score, and consider valid only the alignments that contain U2 introns, i.e., with GT and AG splicing sites. We also skipped sequences with more than ten thousand alignments for computational reasons, and they are probably noise.

3.3.2 SpliceAI

The SpliceAI model [94] was used to compute the probability of each splicing site presented in our pure generations and compared them with the ones in the reference. It is an ensemble of five pre-trained models, all of which are CNNs with diluted convolutions, and it takes a symmetrical context of 10 knt for each base that is evaluated. The model was trained using the GENCODE V24lift37 annotation, with 13 thousand protein-coding genes from which the most conserved isoforms of each gene were included.

The probabilities are computed by following the sample code on their GitHub [95]. That is, by adding 5000 N's downstream and upstream to the sequence, then using one-hot-encoding to a space with (A, C, G, T, N), creating the input matrix $X \in \mathbb{N}^{(L+10000)\times 5}$. We input X to all five models, and the outputs are $Y_m \in \mathbb{R}^{L\times 3} : m \in (1,\ldots,5)$, having for each position three probabilities of being acceptor, donor, or no splicing site. The final probability is the average of all five models $Y = \sum_{m=1}^{5} Y_m/5$.

3.4 Greedy algorithm

The proposed algorithm for the generation of mRNA isoforms, given a gene with sequence S as input, makes use of LoRNA$^{\text{SH}}$ (Sec. 2.3.5) [71], an LLM trained with human and mouse long-read transcripts from 26 cancer cell lines, to work on the probability distribution of the NTP output from the model.

We define the operator $\mathcal{L}$, for LoRNA$^{\text{SH}}$ (Eq. 3.4). With L the dimension of S, which is tokenized to have numerical values, $\bar{S} \in \mathbb{N}^L$, and considering the 32 possible LoRNA$^{\text{SH}}$ tokens. The output is a matrix such that each row corresponds to a position, and columns are the tokens. With $Y = \mathcal{L}(x)$,$Y_{p,t}$ is the probability that at position $p+1$, the token will be t.

$$\mathcal{L} : \mathbb{N}^L \to \mathbb{R}^{L\times 32} \tag{3.4}$$

It is a greedy algorithm because it makes local optimal choices. We further refine our predictions by considering the output of many generations of the algorithm for the same sequence.

3.4.1 The algorithm

LoRNA$^{\text{SH}}$ takes as input a tokenized version of an isoform transcript, where introns and exons are encoded as upper and lower letters, respectively. In addition, there are other tokens for the species (human or mouse), for the DNA flanking region (ACGT $\to$ WXYZ), before the promoter (token S), and after the polyadenylation (token E).

So, given a gene with sequence S, with known start and end points, we start our output sequence x with the human token, 16 flanking DNA tokens, and a start token, as the tokens were used in training. As in the example below, underlined bases are flanking DNA:

$$S = \underline{\text{ACTGCACTTGCCTCGC}}\text{TGCTTCAGTCACGGGGC}\ldots$$
$$x = \text{H}\underline{\text{WXZYXWXZZYXXZXYX}}\text{S}$$

We begin in the exon state. From now on, we iterate over the bases until a GT is found (5'SS) in position p, then we compute the probability that the sequence x would continue as intron or as exon, using LoRNA$^{\text{SH}}$. We create two sequences by expanding x until p by r bases downstream in both conditions: x^{exon} and x^{intron} with probabilities P^e and P^i, respectively. Follow the example below, with $p = 8$, and $r = 6$, considering S the position 0.

$$x = \text{HWXZYXWXZZYXXZXYXStgcttca}$$
$$x^{exon} = \text{HWXZYXWXZZYXXZXYXStgcttca}\underline{\text{gtcacggg}}$$
$$x^{intron} = \text{HWXZYXWXZZYXXZXYXStgcttca}\underline{\text{GTCACGGG}}$$

The probabilities are computed by the chain rule (Eq. 3.5), the dependence on previous bases is intrinsic to the model, and $\bar{x}$ is a tokenized sequence.

$$P = \prod_{i=2}^{p+r-1} \mathcal{L}(\bar{x})_{i-1,\bar{x_i}} \tag{3.5}$$

Since we are comparing P^e to P^i, and all tokens are the same up until position p, we reduce the probability computation (Eq. 3.6). It is a design decision based also on the fact that for large p, the difference in probabilities would be minimal, and the local properties would diminish.

$$P = \prod_{i=p+1}^{p+r-1} \mathcal{L}(\bar{x})_{i-1,\bar{x_i}} \tag{3.6}$$

We then sample the Bernoulli distribution calculated with the softmax of both probabilities. p_k is the computed probability of being an exon or intron, k is the event, and f is the probability mass function.

$$p = \text{Softmax}(P^e, P^i) \tag{3.7}$$

$$f(k; p) = \begin{cases} p^e & \text{if k is exon} \\ p^i = 1 - p^e & \text{if k is intron} \end{cases} \tag{3.8}$$

We assign to x the sequence corresponding to the sampling outcome. E.g. $P^e = 20\%$, $P^i = 90\%$, then $p = (0.33, 0.67)$, if sampling p we get intron, then $x \leftarrow x^{intron}$, and we change the state to intron.

At the intron state, we look for the next AG (3'SS); we do not check for BPs this time because we expect such a check from the model's base knowledge. Once it is found, the same procedure takes place with the following sequence (Fig. 3.7).

$$\begin{aligned} x &= \text{H}\ldots\text{StgcttcaGTCACGGG}\ldots\text{CGTCGAG} \\ x^{exon} &= \text{H}\ldots\text{StgcttcaGTCACGGG}\ldots\text{CGTCGAGtgctac} \\ x^{intron} &= \text{H}\ldots\text{StgcttcaGTCACGGG}\ldots\text{CGTCGAGTGCTAC} \end{aligned}$$

We iterate until the sequence ends, insert the polyadenylation token E, add the downstream flanking DNA, and a [SEP] token at the end. However, polyadenylation occurs at the mRNA, so it cannot take place on introns. So, the token E is inserted immediately after the latest exon base of x, and the downstream intron tokens are translated to flanking DNA (Fig. 3.8).

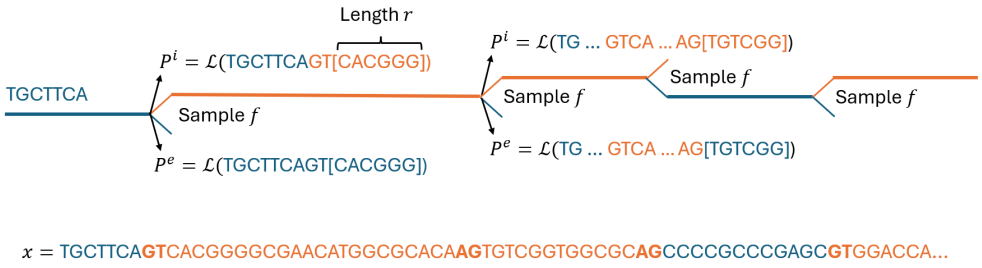

Figure 3.7: Greedy algorithm, some iterations. Only introns and exons are depicted for simplicity. Blue represents exons, and orange represents introns.

In the development phase, a 1000 isoforms with variable promoters were generated for the genes FOXA1, MYC, HRAS, with hyperparameter r taking values 6, 8, 12, 15, 20, 50, 75, 100, 150, and 200. The position of the possible promoters originates from the IIGM's dataset.

3.4.2 Perplexity of the algorithm

Perplexity is a measure of uncertainty when sampling from a discrete probability distribution (Eq. 3.9). It was created in the context of speech recognition tests [96], and it is the exponential of the entropy of a probability distribution.

A probability distribution f with $PP(f) = m$ has the same uncertainty as a fair m-sided dice. We compare the discrete probability distribution f with 2 choices, so $1 \leq PP(f) \leq 2$.

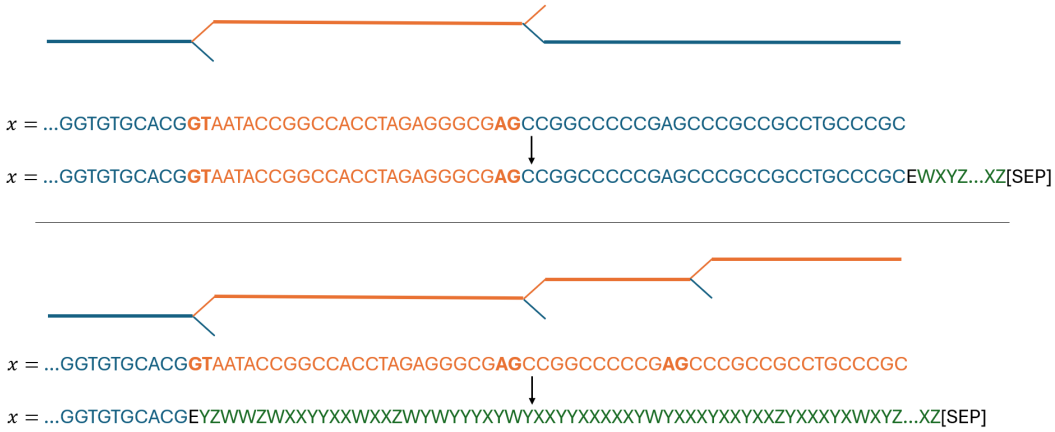

Figure 3.8: Final step of the greedy algorithm. Top: final token of x is an exon. Bottom: final token is an intron. Blue represents exons, orange introns, green flanking DNA, and black special tokens.

$$PP(p) = \prod_x p(x)^{-p(x)} = b^{-\sum_x p(x)\log_b p(x)} \tag{3.9}$$

We want to evaluate the certainty of the choices of our algorithm. Two types of distributions are presented, the first are for each decision the algorithm makes, therefore, for each splice signal it finds, and the second are the distributions when it changes state from intron to exon and vice-versa. We plot the information categorized by r and by gene.

3.4.3 Counting the created isoforms

We count the number of times a certain isoform was generated for each r. We observed a low count due to randomness in the selection of the same GT or AG subsequence. Hence, we created a method to take these errors into account.

We define the error variable e, with ranging values 5, 10, 20, and 50 nt. Given an e, we create a mesh of points that merge e bases. So each isoform is translated into these meshes, and we count them instead of the pure sequence. The translation is such that, in the original sequence x, if there is any exon token inside the interval $[x_{i \mod e}, x_{i \mod e+e})$, the mesh point corresponding to that interval takes value 1, else 0 (Fig. 3.9).

Figure 3.9: Mesh for counting generated isoforms with $e = 5$. x the original sequence, with lower letters in yellow boxes as exons, and upper letters in purple boxes as introns. And x^m is the mesh created from the original sequence, with the same color pattern, and 1 meaning exon, and 0 meaning intron.

3.4.4 Distribution of Exon expression

For each gene and each promoter used as the initial point of generation, we count the number of times a base is expressed as an exon; therefore, expressed in the mRNA isoform. The counts are normalized to the total number of generations in which such a base is part of the isoform, and hence, not defined in flanking DNA tokens, following the expression in Eq. 3.10, with N the number of isoforms generated from the same promoter site. An example is provided at the top of Fig. 3.10.

$$C_i = \sum_{g=0}^{N} \frac{\mathbf{I}(x_i^g \text{ is exon})}{\mathbf{I}(x_i^g \text{ is intron} \vee \text{exon})} \tag{3.10}$$

We also produced a correlation matrix of exon expression to be able to evaluate AS events, such as cassette and mutually exclusive exons.

3.4.5 Computational analysis

While running the greedy algorithm, we recorded the elapsed time, CPU time, GPU time, and peak VRAM of 25000 generations, 250 for each isoform. It was performed on the test dataset, so the maximum length of the genes was 10 knt.

3.5 Base-level Binary Classifier

Each base will have an exon expression score C_i from 0 to 1. With a varying threshold t, we can classify base-by-base exons from introns (Fig. 3.10).

3.5.1 Validation

We evaluated our algorithm with the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) [97], which is a curve plotted on two axes, $1 -$ Specificity $\times$ Sensitivity or FPR $\times$ TPR, see Eq. 3.12, 3.11 and Fig. 3.11. The points on the curve are obtained by evaluating our algorithm (Sec. 3.5) with threshold t

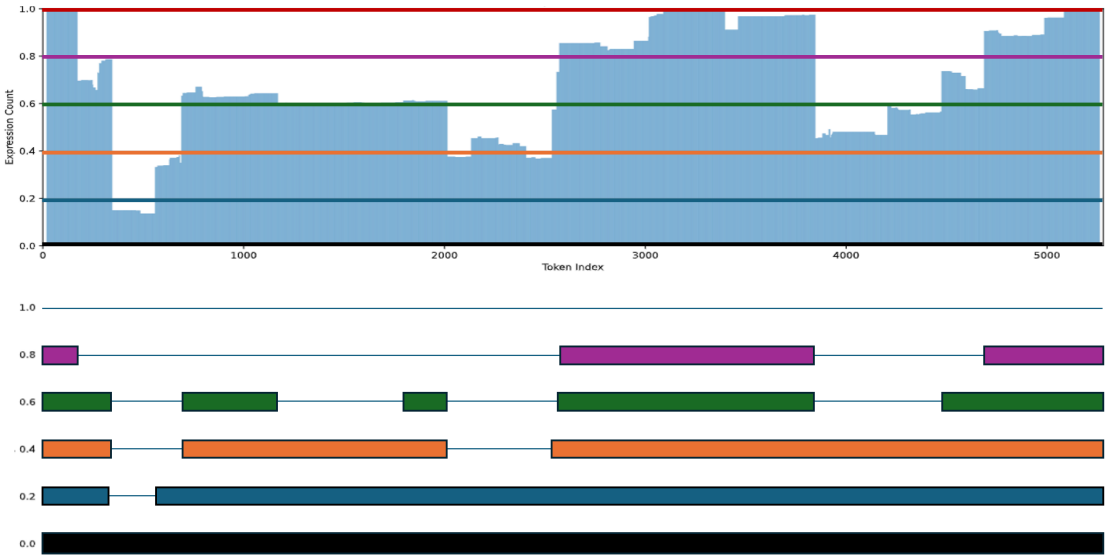

Figure 3.10: Example of our exon classifier, for $t = 0$, 20%, 40%, 60%, 80%, and 100%. At the top, the expression of each base, with a line for each threshold. At the bottom, the produced isoforms. Colors represent each t.

varying from 0 to 1, taking steps of 2%. Each evaluation is computed base by base, so given a known isoform, we computed N isoforms with our algorithm with the same promoter; we consider the classification of individual bases as exons (positive) or introns (negative). We count the sum across all bases of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) to compute the specificity and sensitivity of our classifier.

$$1 - \text{Specificity} = \text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{3.11}$$

$$\text{Sensitivity} = \text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.12}$$

A classifier with AUC $= 0.5$ is as good as random guessing, or flipping a coin. The greater the area, the better a classifier is, for it can separate well TPs and FPs, reducing both errors of type-I and type-II. Finally, a perfect classifier has AUC $= 1$, where, $\exists t : (\text{TPR} = 1, \text{FPR} = 0)$.

We computed the average AUC for five genes, with various r as validation. Also, the average AUC weighted by expression is computed to give more importance to the mRNA isoforms that are more commonly observed. Table 3.5 presents a proposed interpretation of the AUC values in diagnostic accuracy studies [99].

Finally, to test our algorithm, we generated 257 isoforms for 51 different genes, with multiple known promoters, and computed the AUC.

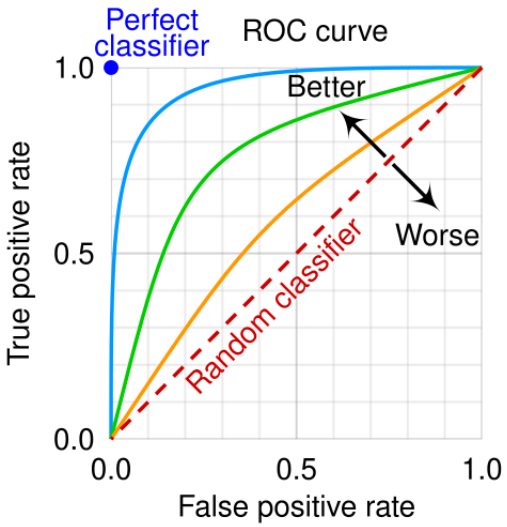

Figure 3.11: Example of ROC curve. Originally from `Wikipedia` [98] (CC BY-SA 4.0)

AUC value	Interpretation
$0.9 \leq$ AUC	Excellent
$0.8 \leq$ AUC < 0.9	Considerable
$0.7 \leq$ AUC < 0.8	Fair
$0.6 \leq$ AUC < 0.7	Poor
$0.5 \leq$ AUC < 0.6	Fail

Table 3.5: Interpretation of AUC values

Chapter 4

Results

4.1 Generation of isoform with splicing signals

In the first attempt at generating the possible isoforms from a gene, we first consider splitting at each U2 splice site, i.e., GT and AG. Given two randomly chosen genes, i.e., PLRG1 and FUS, we notice that the computational cost in terms of time and memory is unfeasible, e.g., $\sim 10^{556}$ splits. In our attempts at reducing the number of splits, we computed the probability of each motif around the splicing sites (Sec. 3.2.1), and would split only at splicing sites that were ranked in the top T motifs in terms of probability. We notice that for low T, no results match real isoforms, and for high T, it becomes unfeasible again. Information regarding the computational costs of the first approach and the one ranking splicing site motifs is presented in Tab. 4.1. Additional measures were taken to reduce the actual computational time, not the worst case, i.e., considering a regular expression for BP motifs and their distance from the 3'SS, but such efforts were in vain, since there was no positive outcome.

Furthermore, we notice that in four real isoforms of the gene FUS, the ranking of the observed motifs reaches up to $T = 3225$ (Tab. 4.2), meaning that for any smaller T, no valid isoform would ever be generated by this method. A conclusion that was confirmed by our results, that after hours of computing and many GB of saved sequence data, no generated isoform matched any real one; even though we stopped the computations due to the high cost, we still could not expect anything better from additional time and memory consumption.

This leads to the conclusion that we need a more sophisticated method for generating possible isoforms, given a gene. We will use LoRNA$^{\text{SH}}$ within a greedy algorithm, whose generations will be used inside a base-by-base exon classifier.

Name	PLRG1	FUS
Gene ID	ENSG00000171566	ENSG00000089280
Length (nt)	5930	11455
Isoforms	2	4
Max exons	15	15
GT count	914	986
AG count	934	1049
Worst case splits number	$\sim 10^{556}$	$\sim 10^{612}$
Top 10 5'SS count	2	2
Top 10 3'SS count	5	3
Worst case splits number	128	32
Top 50 5'SS count	6	11
Top 50 3'SS count	15	18
Worst case splits number	$\sim$ 16 M	$\sim$ 536 M
Top 100 5'SS count	9	16
Top 100 3'SS count	20	41
Worst case splits number	$\sim$ 536 M	$\sim 10^{17}$
Top 500 5'SS count	29	55
Top 500 3'SS count	132	142
Worst case splits number	$\sim 10^{18}$	$\sim 10^{59}$

Table 4.1: Information of two sample genes, counts of splicing signals, and motifs within certain ranks of probabilities. The number of splits necessary to generate all isoforms is calculated in accordance with Eq. 3.2.

	List of ranks of observed motifs in all four FUS isoforms
5'SS	4, 8, 40, 45, 49, 65, 143, 184, 233, 272, 401, 466, 562, 1871, 2398
3'SS	57, 78, 102, 170, 416, 465, 527, 914, 989, 1177, 1200, 1300, 2051, 2873,3225

Table 4.2: List of ranks of observed motifs in FUS gene.

4.2 Reproducing LoRNA^{SH} Results

As described in Sec. 3.3, we have generated a thousand isoforms with LoRNA^{SH}'s NTP, using as initial prompt the human species and the start token, HS; the

maximum length of each pure generation is set to 20 knt. In this section, we will use BLAT to compare our pure generations with the reference genome GRCh37; compare the statistics on the lengths of introns, exons, sequences, and mRNAs between our experiments and IIGM's dataset; Use SpliceAI to calculate the probability of the used splicing sites ot being so, still comparing against IIGM's dataset; and finally, attempt to correlate the long-read expressions with the End token and normalized sequence probabilities, as it is done in the original paper of the model [71].

4.2.1 BLAT

We have fed as input to the BLAT algorithm (Sec. 3.1.2) the pure generations in two different ways, either by using the gene sequence (introns and exons), or only the exons (mRNA). The matched bases, in absolute values and in relative terms to the sequence lengths, are presented in Fig. 4.1. We can conclude that LoRNA^{SH} does not produce real genes or isoforms by heart, i.e., it is not overfit to predict the next tokens of the sequences seen in the training set.

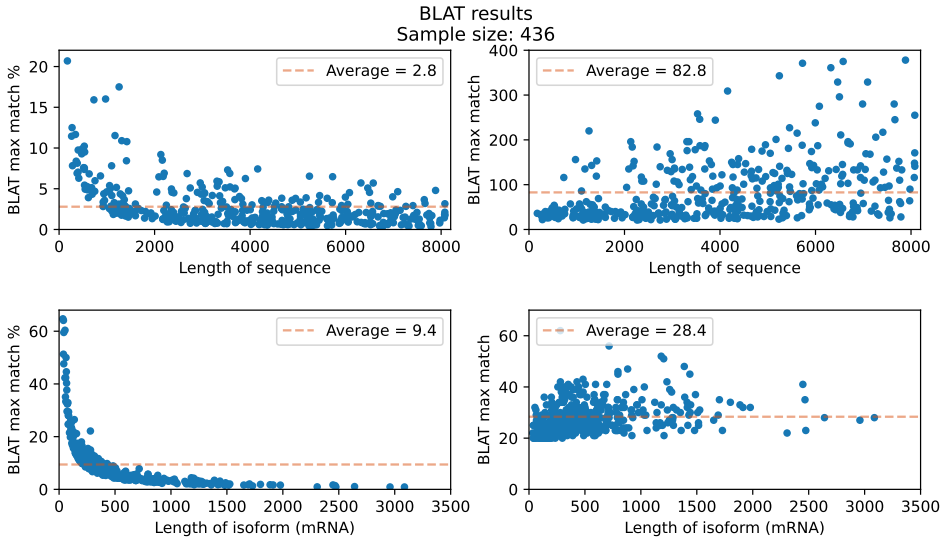

Figure 4.1: BLAT results from pure generations. On the Left, maximum matches normalized by the transcript length; on the Right, maximum matches in absolute values; on Top, data describing the BLAT of the generated sequences, i.e., flanking DNA, introns, and exons; at the Bottom, data of mRNA, i.e., only exons. Sample size is smaller due to not finding matches ≥ 20, and to the size limit of GET requests, i.e., 8KB.

4.2.2 Introns and Exons Statistics

We can compare the distributions of counters and lengths of introns and exons between our pure generations, IIGM's dataset, and the distributions presented in the original paper [71] in Fig. 4.2. Details on the processing are available in Sec. 3.3.1. Comparing our distributions and theirs, there is no direct match; this might be due to different post-processing of the data.

Analysing in detail Fig. 4.4, we produce longer exons and introns with respect to the reference (Sec. 3.1.1). Whilst producing shorter sequences and mRNAs, which makes sense with the fact that LoRNA^{SH} produced fewer exons and introns than the reference (Fig. 4.3), this might be due to a bias towards staying in the same state (intron/exon) as it computes the probabilities of the next token. The distributions of the original paper exhibit similar trends.

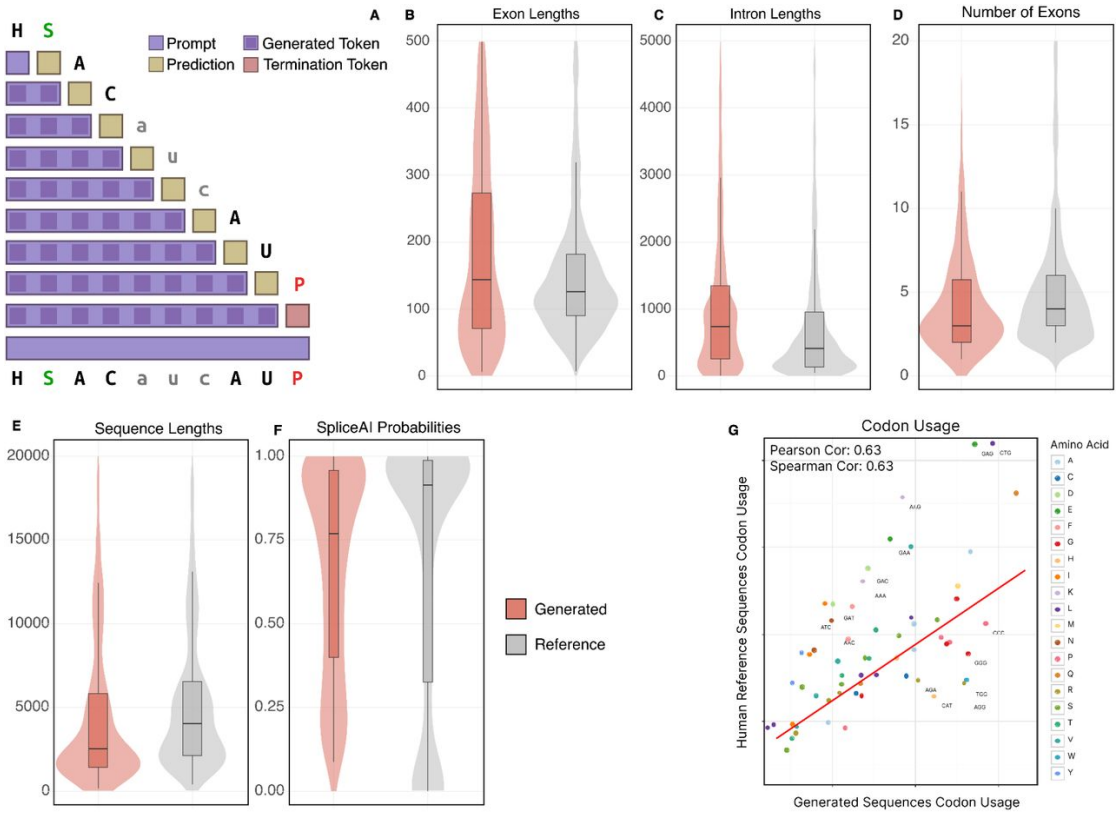

Figure 4.2: Distribution of pure generations statistics from original paper. A: Depiction of the pure LoRNA^{SH} generations with initial prompt HS, up to the last token P, for polyadenylation, which in our case is called E, for end token. B: Exon lengths. C: Intron lengths. D: Number of exons. E: Sequence lengths. F: Splice probabilities. G: Correlation of the Codon usage between their Human reference and their pure generations. Original from [71] (CC-BY-NC-ND 4.0)

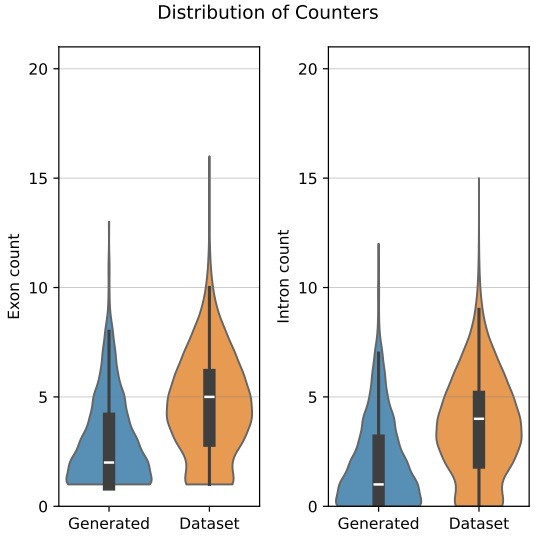

Figure 4.3: Distribution of exon and intron counters in pure generations and reference. Our pure generations are presented in blue, and IIGM's data, in orange. For sample sizes, refer to Tab. 4.3.

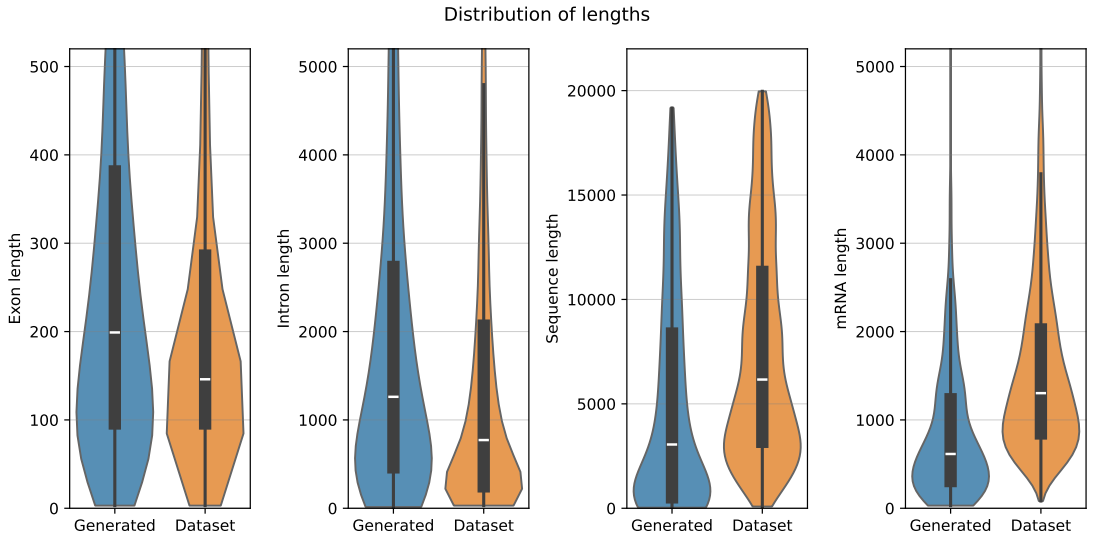

Figure 4.4: Distribution of exon, intron, sequence, and mRNA lengths in pure generations and reference. Our pure generations are presented in blue, and IIGM's data, in orange. For sample sizes, refer to Tab. 4.3.

	Pure generations	IIGM's dataset
Analyzed isoforms	836	10336
Total number of exons	2532	47943
Total number of introns	1696	37607
Excluded (small)	2	0
Excluded (big)	195	7527
Excluded (big RNA)	5	0
Excluded (full intron)	76	0
Total noise tokens	738	0
Total tokens	4.146.229	77.712.590

Table 4.3: Sample sizes of pure generations and IIGM Statistics. Noise tokens are not included in the Total tokens value, nor are the excluded isoforms data used in any way. Details on processing in Sec. 3.3.1.

4.2.3 SpliceAI on generated Splicing Sites

Using the SpliceAI model to evaluate the probability of each splicing site on both our reference and on the pure generations, we notice an important discrepancy (Fig. 4.5) with the original paper (Fig. 4.2); such a difference could be due to different data-processing, or perhaps even how the SpliceAI model was employed, our methods are explained in Sec. 3.3.2. Comparing our probabilities, we notice the reference probabilities are much higher compared to the pure generations; the former has a median of 99.03% and the latter 0.23%.

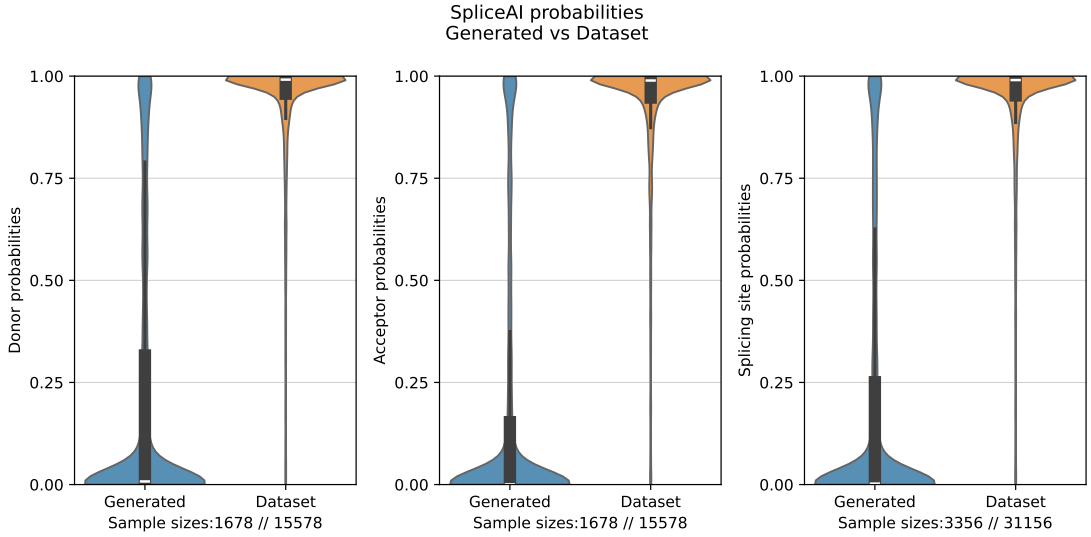

Figure 4.5: Distribution of SpliceAI probabilities in pure generations and reference. Our pure generations are presented in blue, and IIGM's data, in orange. The represented sample sizes at the bottom refer to the number of splicing sites.

4.2.4 Expression vs Full sequence and End token probabilities

The original paper correlated the long-read expression with the E token probability, yielding a Pearson Correlation Coefficient (PCC) of 0.15 and a p-value of $8.6e^{-237}$ [71]. However, we did not obtain these results, as we can observe in Fig. 4.6 in the middle. In the same figure, we compute the Pearson and Spearman correlations between the model probabilities and these against the long-read expression, and none present any statistically significant correlation. Additional tests were performed by considering the natural logarithm of either or both axis in all three graphs, but no significant improvement was observed. Given these results, these probability metrics will not be used as a fitness metric of any kind for future sections.

An interesting observation in Fig. 4.6, the main normalized probability is about 30%, and some sequences had it greater than 40%; We can take in regard 8 main tokens (A, C, G, T, a, c, g, t), but it is plausible to consider that the main probabilities are between tokens of the same kind, between exon and intron; therefore, given our semplification, a complete random next token generator would have the normalized probability constant at 25%. No meaning was extracted from the probability distribution of the E token.

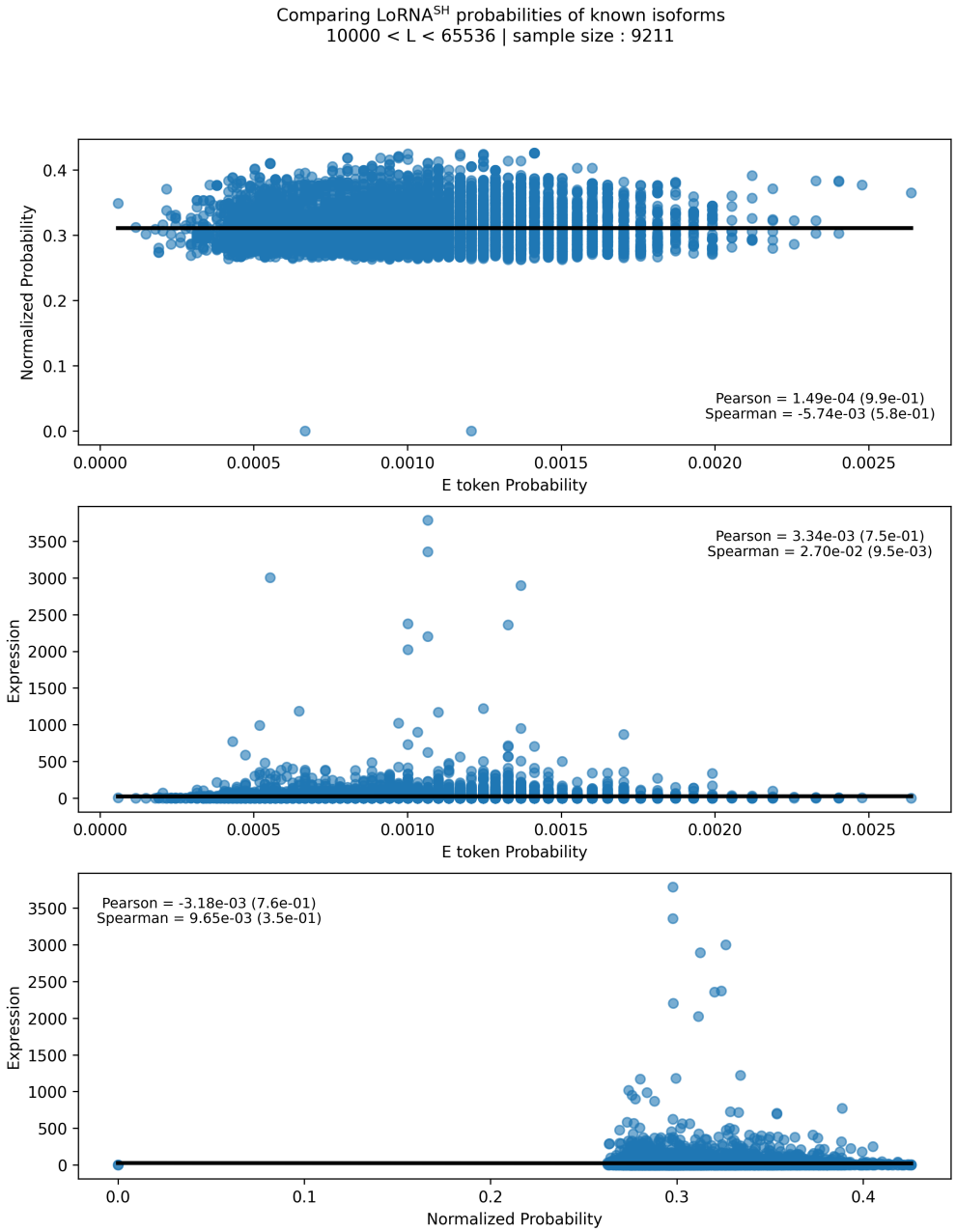

Figure 4.6: Model probabilities and Dataset Expression. On Top, the E token probability vs the normalized probability of the whole sequence; in the Middle, the E token probability vs the long-read expression; at the Bottom, the normalized probability of the whole sequence vs the long-read expression. In parentheses, the p-value of each correlation coefficient. The black lines represent the linear regressions.

4.3 Greedy algorithm

We will first study our algorithm with the genes FOXA1, HRAS, MYC, KLF6, MDM2, and SRSF1, whose information is in Table 4.4; all information is obtained from the readings of the IIGM dataset. A first example of some of the generated isoforms compared with known GRCh37 isoforms is presented in Fig. 4.7 with the BLAT alignment of the UCSC genome browser. The algorithm is described in Sec. 3.4.

Gene	Iso ID	Exp	Seq Length	# Exons	Avg Exon Length	Category
FOXA1	2479a	159.5	5241	2	1526	known
FOXA1	e15ae	5.3	5241	1	5241	novel
FOXA1	33e8f	1.3	5241	3	1051	known
FOXA1	e0a58	1.2	784	2	248	known
HRAS	b87f8	30.7	3309	6	174	known
HRAS	c833c	15.2	3098	6	165	novel
HRAS	5bf6c	13.9	3309	6	206	known
HRAS	11874	2.7	3309	7	161	known
HRAS	bc9d7	1.0	582	1	582	novel
MYC	f32d7	227.4	5204	3	720	novel
MYC	a957b	223.7	5351	3	784	known
KLF6	367bb	26.9	8641	3	1298	known
KLF6	90cc4	5.7	8641	4	1004	known
KLF6	a5a20	0.6	5736	1	5736	novel
KLF6	06a0a	0.5	1541	1	1541	others
MDM2	8e5f4	10.9	1760	1	1760	novel
SRSF1	4d51f	94.7	3781	4	699	known
SRSF1	97e76	28.2	3175	1	3175	novel
SRSF1	06e6c	19.2	3781	5	263	novel
SRSF1	bf922	5.4	3781	4	519	known
SRSF1	145cd	3.6	3781	5	375	novel

Table 4.4: Information on 21 isoforms present in the IIGM dataset, from 6 genes. *Iso ID* refers to the unique sequencing identifier. *Exp* is the long-read expression in TPM. *Seq Length* is the distance from the first and last exon bases when aligned to the reference GRCh37. The isoform 06a0a is categorized as *others* because it is probably a pre-mRNA fragment.

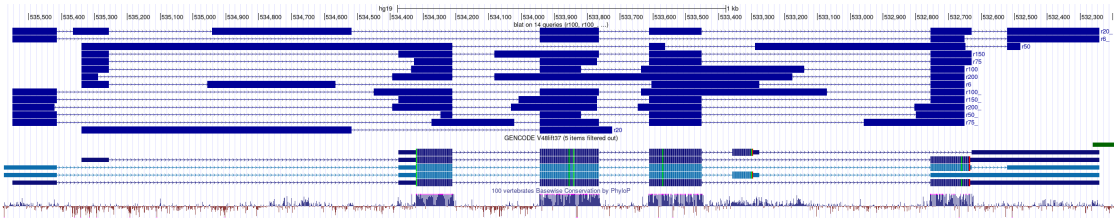

Figure 4.7: Examples of greedy-generated isoforms in UCSC - HRAS. Two different promoters are shown, with variable hyperparameter r, displayed on the right of each alignment.

For each gene, considering only the first promoter, the 10 most frequent isoforms from 200 generations are presented in Figures 4.8-4.13, with an error range of 10 nt and the hyperparameter $r = 75$ (Sec. 3.4.3); the same figures contain isoforms from the IIGM dataset with the TPM long-read expression on the top.

Figure 4.8: Top 10 greedy and IIGM isoforms - FOXA1. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

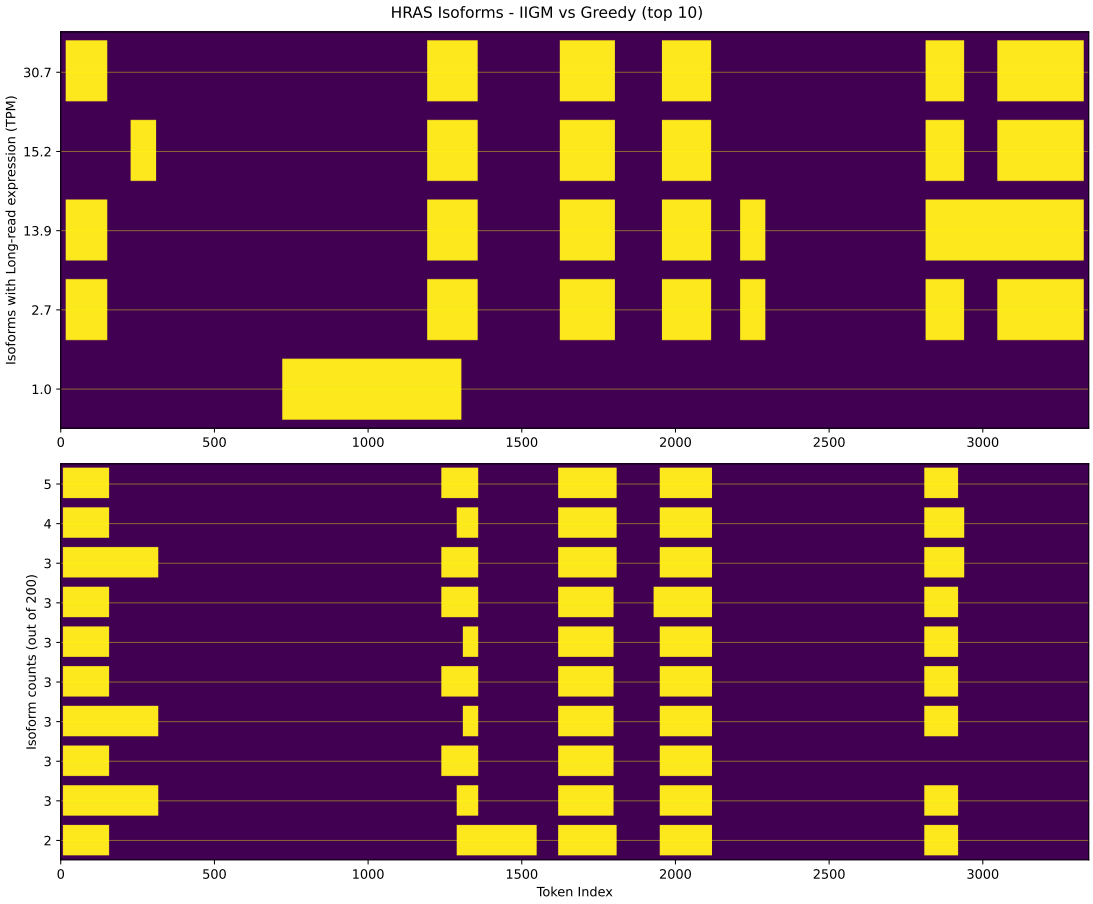

Figure 4.9: Top 10 greedy and IIGM isoforms - HRAS. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

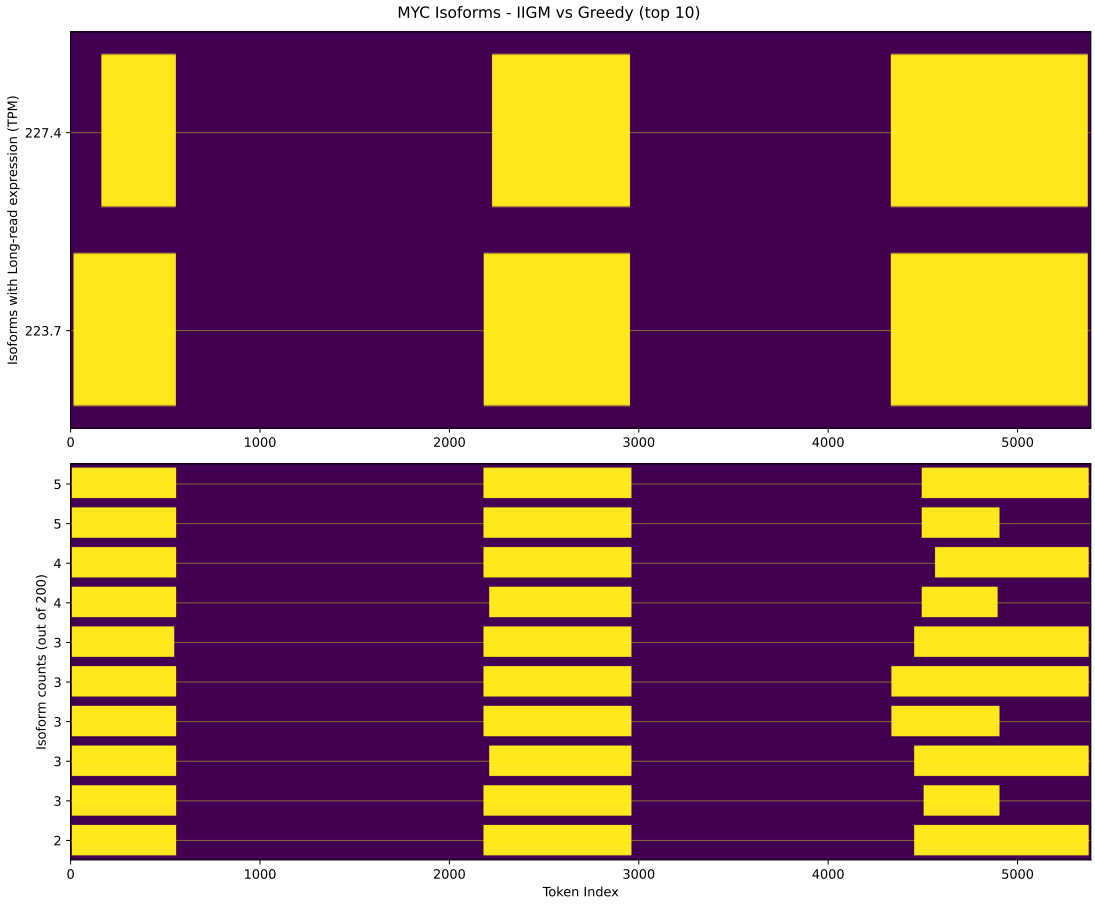

Figure 4.10: Top 10 greedy and IIGM isoforms - MYC. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

Figure 4.11: Top 10 greedy and IIGM isoforms - KLF6. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

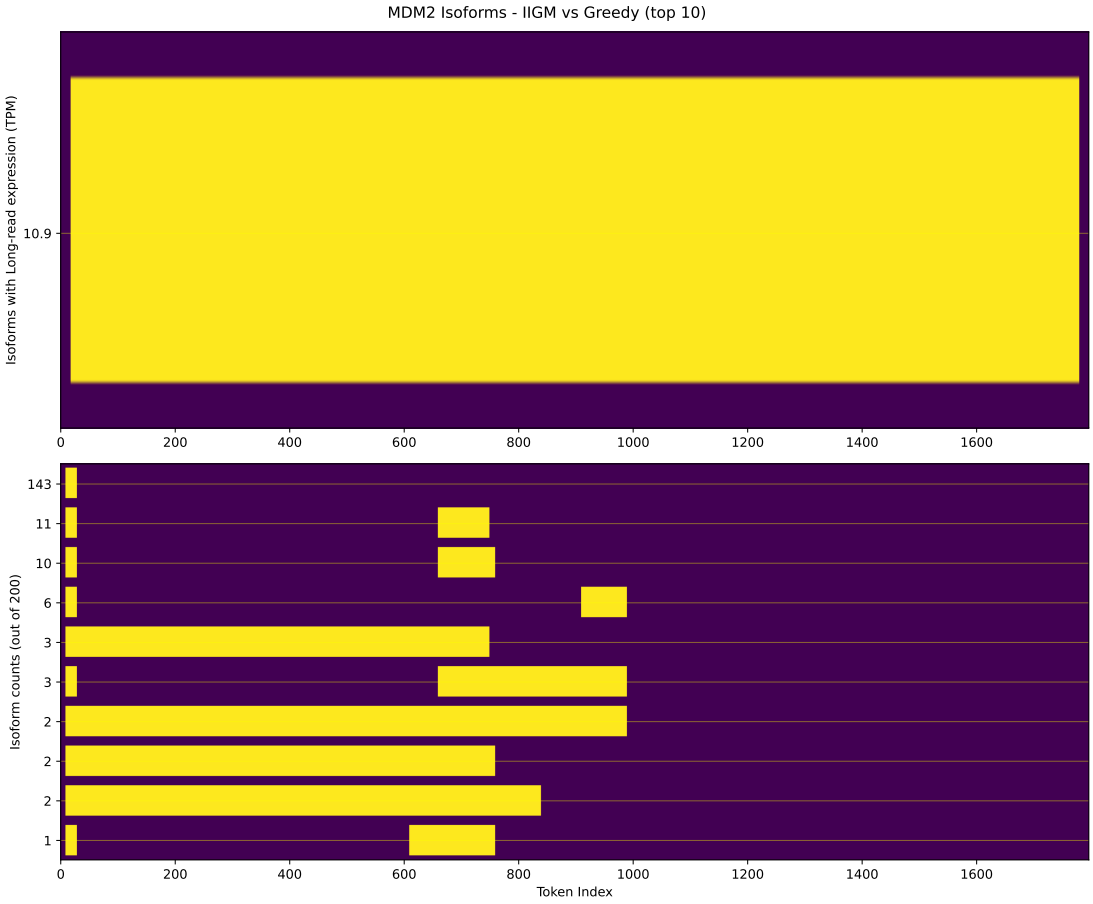

Figure 4.12: Top 10 greedy and IIGM isoforms - MDM2. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

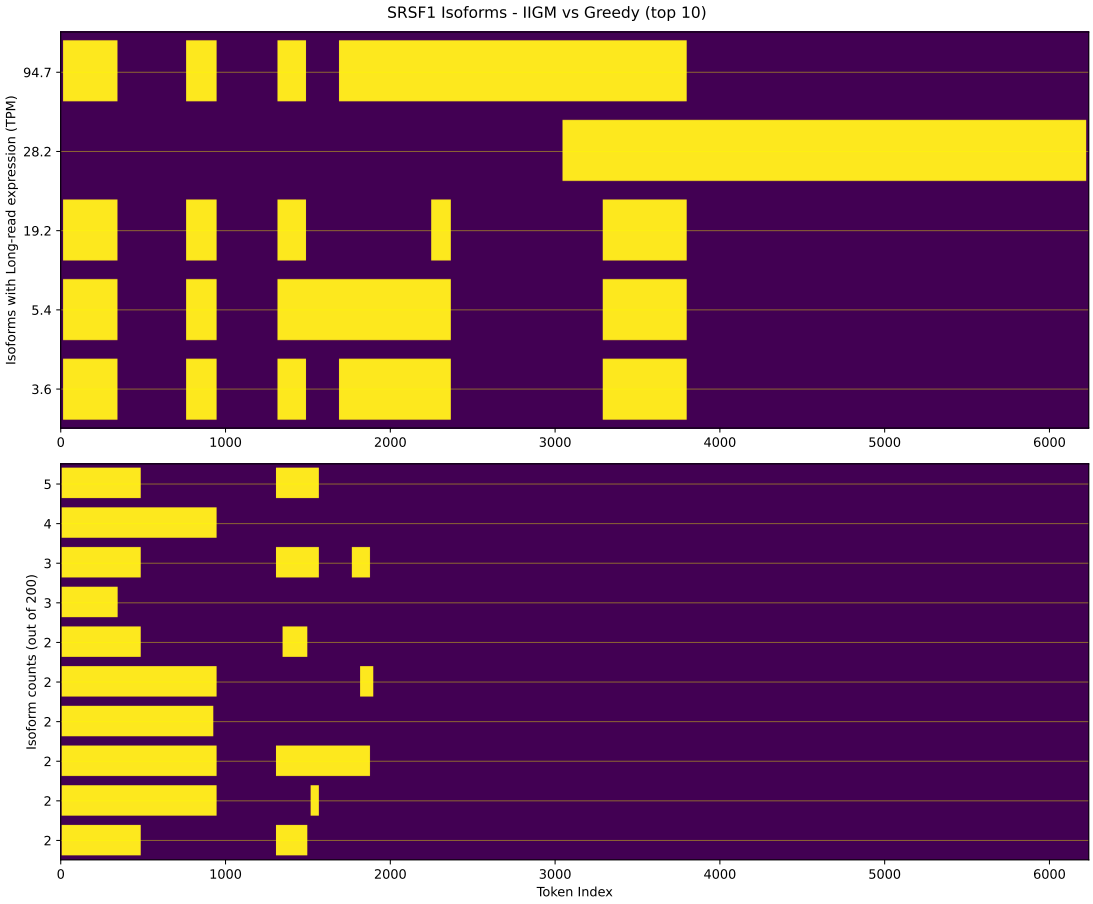

Figure 4.13: Top 10 greedy and IIGM isoforms - SRSF1. Yellow represents exons. On top, the dataset's isoforms with long-read expression in TPM. At the bottom, the 10 most frequent isoforms in the greedy algorithm with an error range of 10 nt and $r = 75$.

4.3.1 Exon expression in Greedy generations

In Figures 4.14-4.19, we plot the exon expression of the genes FOXA1, HRAS, MYC, KLF6, MDM2, and SRSF1 (Tab. 4.4), as described in Sec. 3.4.4, with 200 generations for each promoter site present in IIGM's dataset, and for $r = (6, 20, 50, 75, 100, 150, 200)$. On top of each figure, the isoforms from GRCh37 are depicted with the UCSC genome browser, so that we can compare the global picture of our generated isoforms with the known ones. A particular case is the MDM2 gene, whose only region inside the IIGM's dataset is around five thousand bases away from the nearest MDM2 coding region.

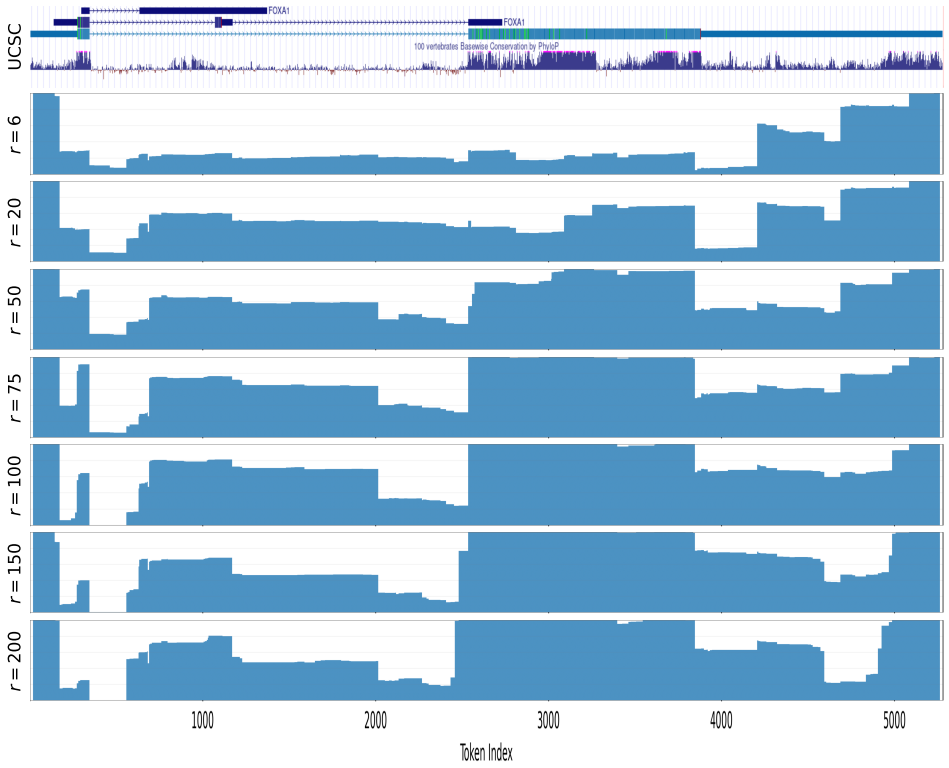

Figure 4.14: Exon expression of greedy generations - FOXA1. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 400 greedy generations with variable r.

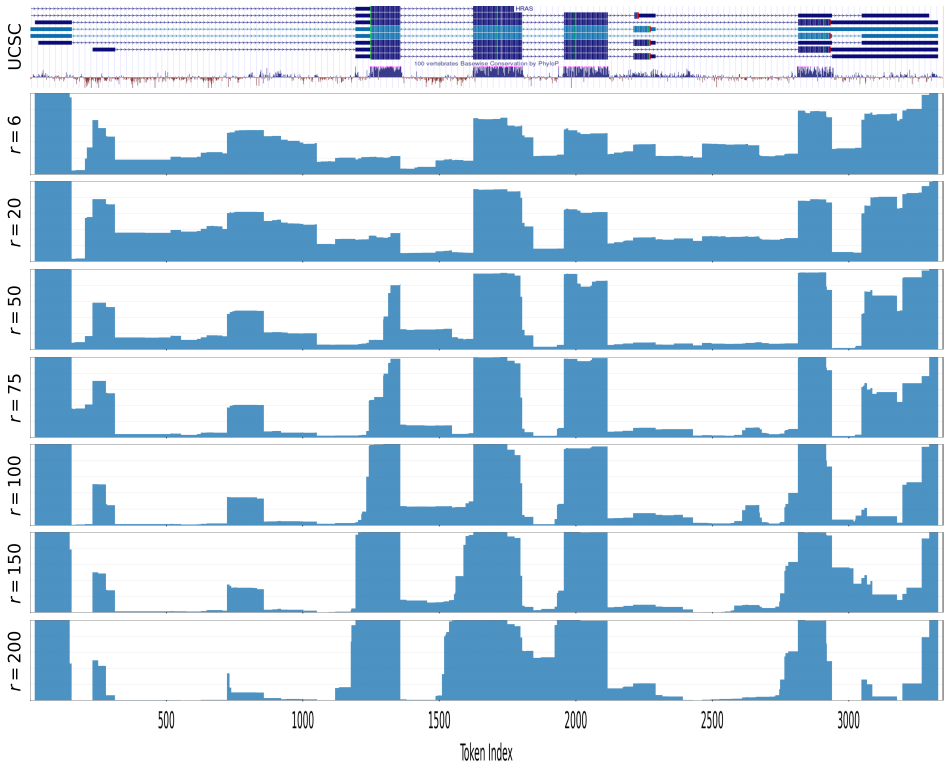

Figure 4.15: Exon expression of greedy generations - HRAS. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 600 greedy generations with variable r.

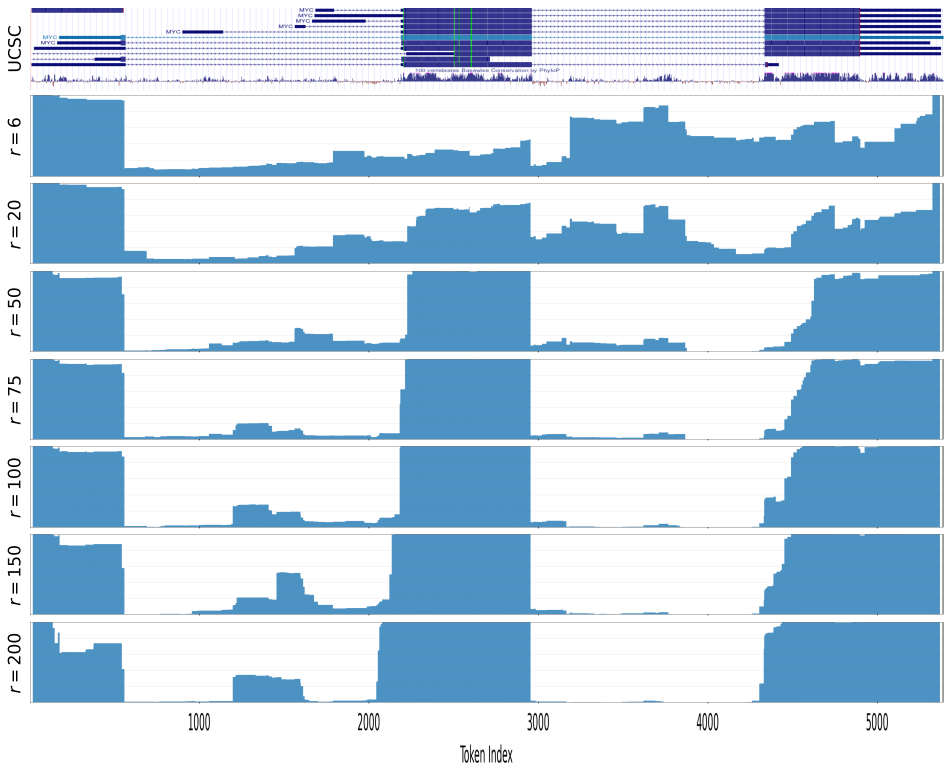

Figure 4.16: Exon expression of greedy generations - MYC. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 400 greedy generations with variable r.

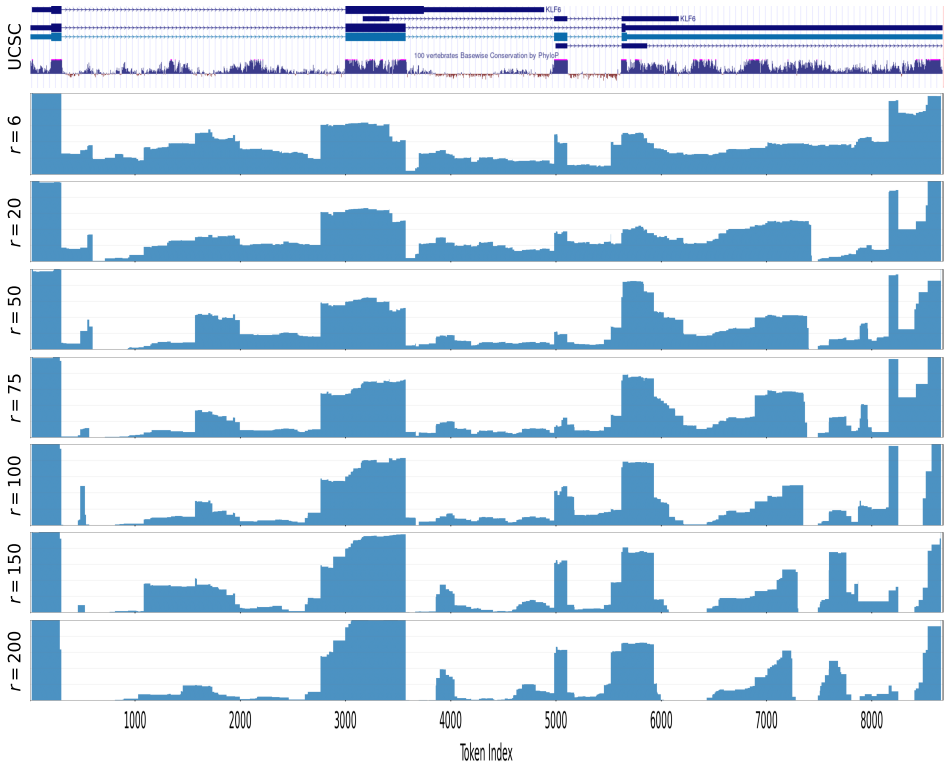

Figure 4.17: Exon expression of greedy generations - KLF6. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 400 greedy generations with variable r.

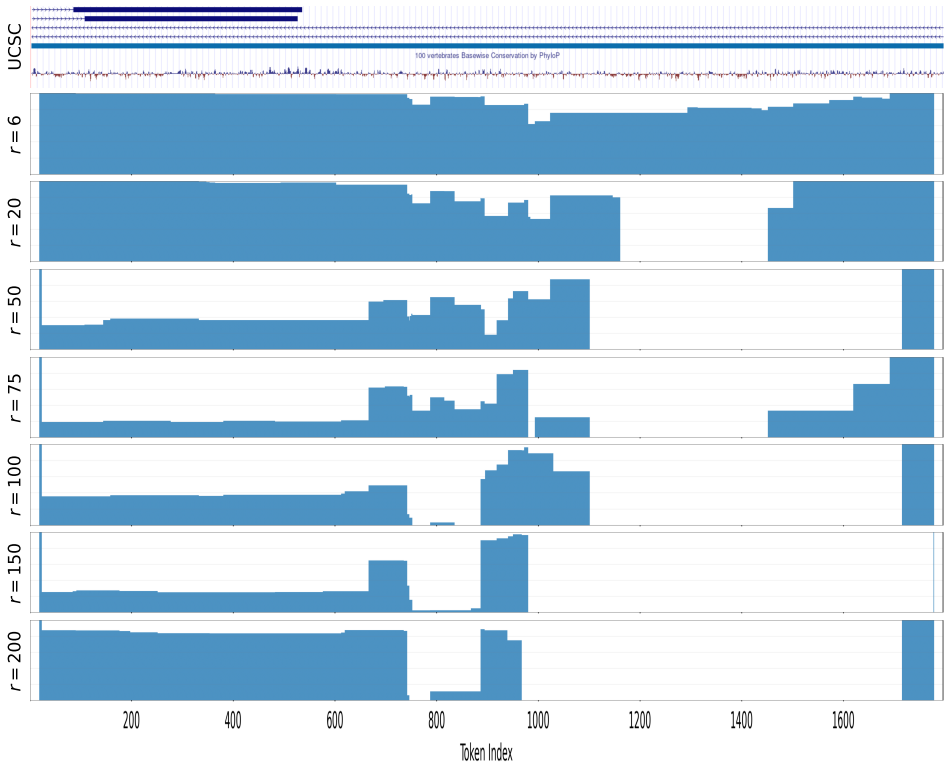

Figure 4.18: Exon expression of greedy generations - MDM2. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 200 greedy generations with variable r.

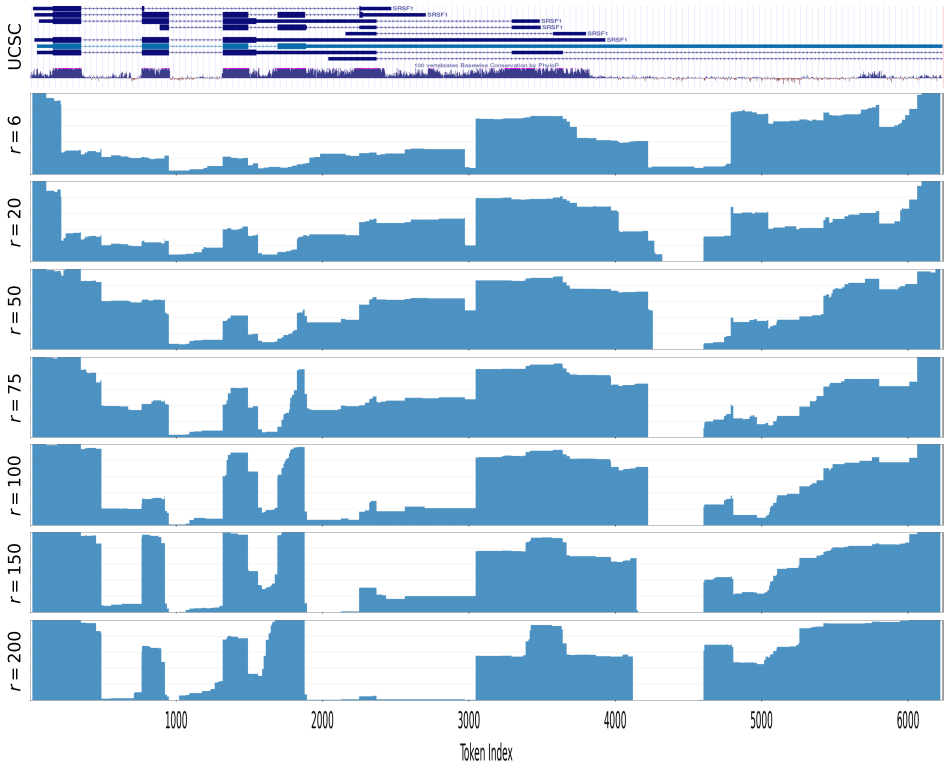

Figure 4.19: Exon expression of greedy generations - SRSF1. On top, GRCh37 isoforms from the UCSC genome browser. Below, the exon expression of 200 greedy generations with variable r.

In the correlation matrices of exon expression presented in Figures 4.20-4.25, we can visualize diagonal blocks that are highly correlated, representing introns and exons within greedy generations. Highly correlated blocks outside the diagonal, e.g., (i, j), means that block i is classified as an exon or an intron frequently when the j block is also classified as the first.

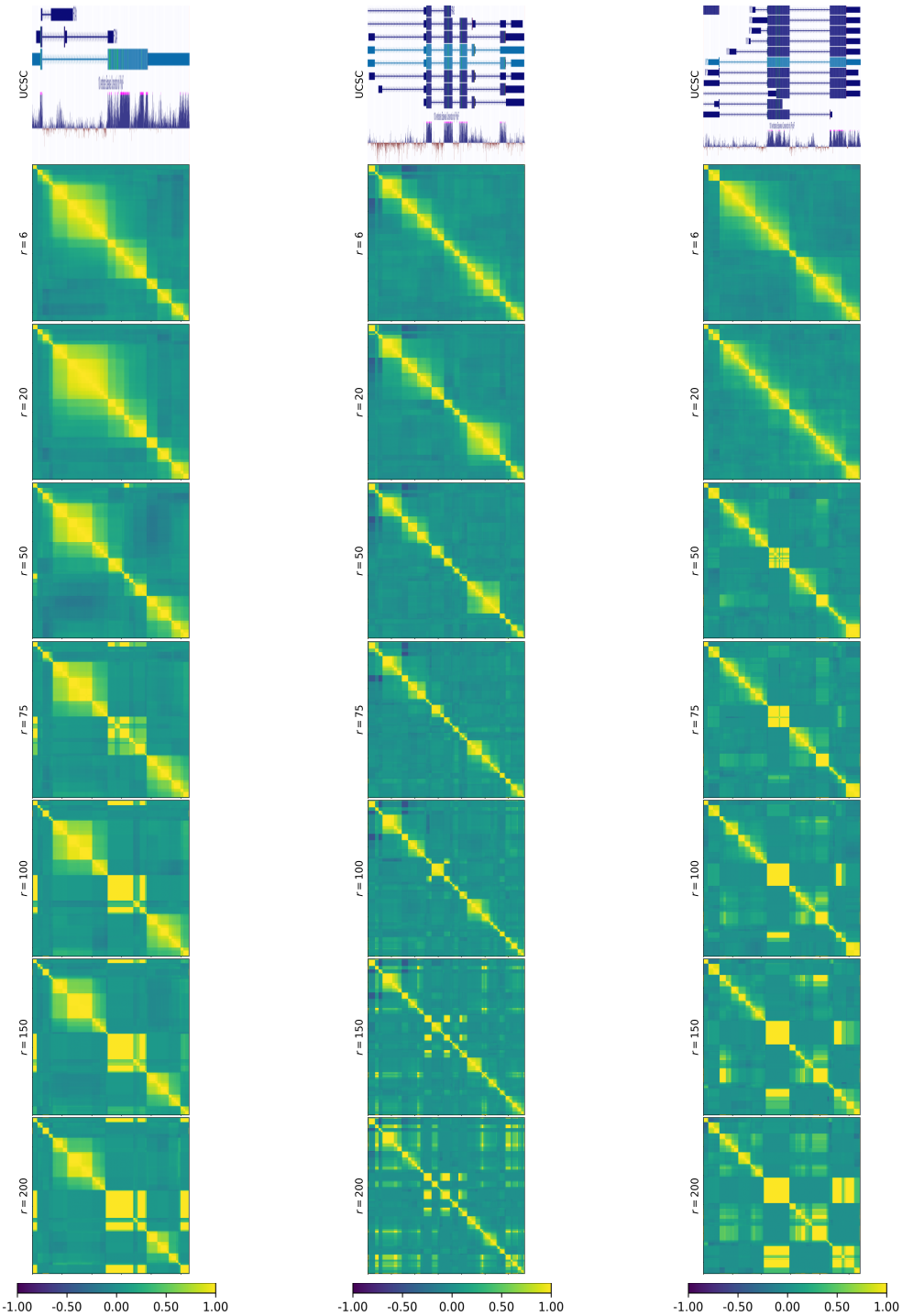

Figure 4.20: Correlation of Exon/Intron expression - FOXA1

Figure 4.21: Correlation of Exon/Intron expression - HRAS

Figure 4.22: Correlation of Exon/Intron expression - MYC

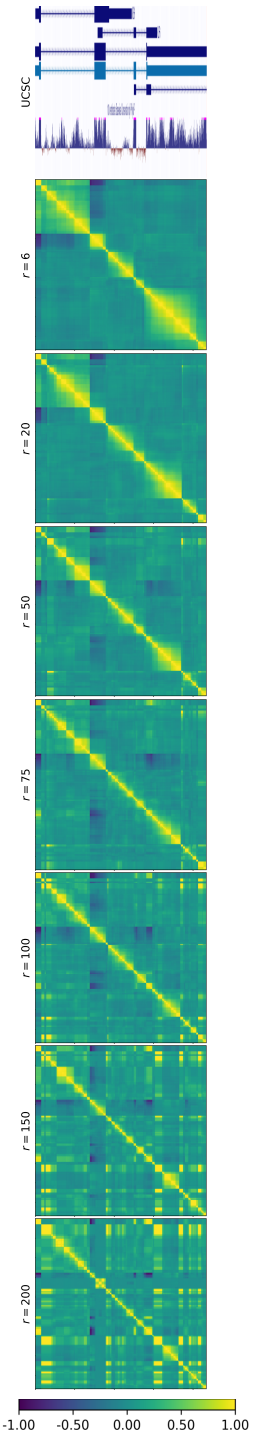

Figure 4.23: Correlation of Exon/Intron expression - KLF6

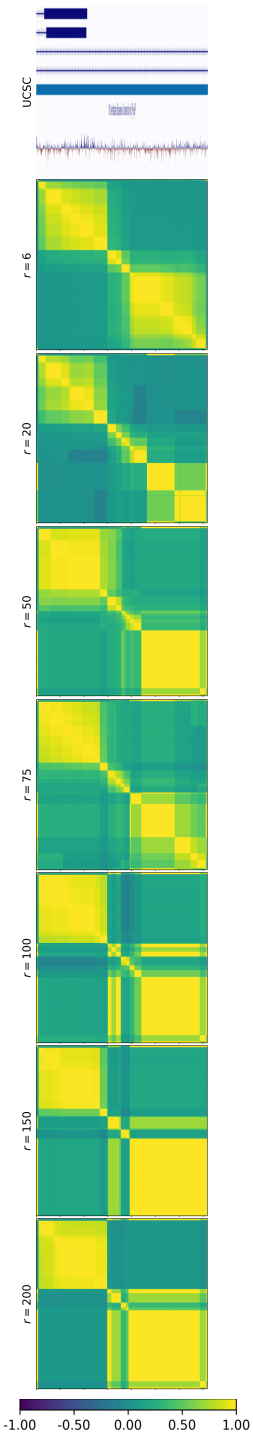

Figure 4.24: Correlation of Exon/Intron expression - MDM2

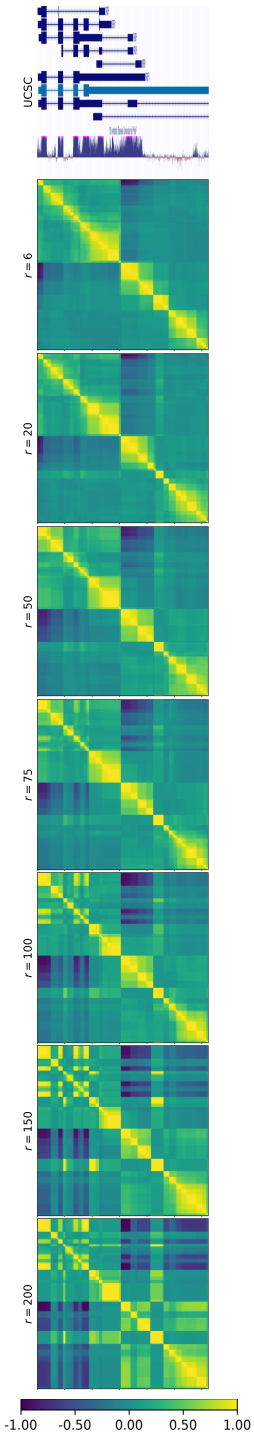

Figure 4.25: Correlation of Exon/Intron expression - SRSF1

With both global and individual exon expression plots and their correlation matrices, we analyze the overall results and discuss the hyperparameter r.

In FOXA1 (Fig. 4.8, 4.14, 4.20), the beginning of the first intron is completely recognized by $r \geq 100$, but all values of r obtained low exon expression in that area. In the first exon, we notice that for the generations whose promoters are positioned upstream to the coding region, the non-coding region is lowly expressed for $r \geq 75$. With $r \geq 50$, there is an overall good recognition of the beginning of the second exon. For all r, there is a drop in expression within the last exon right at the stop codon, i.e., the end of the coding region. As we will notice in most genes, $r = 6$ and $r = 20$ perform worse, and the higher the r, the further our context window includes real exons, making it such that the output probability of the sequence ahead of the splicing site being evaluated is an exon be higher than with lower r, so exons tend to begin earlier than they should, in this gene, this can be seen for both the second exon of the first isoform from UCSC, and the second exon of the third isoform. So, we need to balance this hyperparameter well. Furthermore, around the second exon of the second UCSC isoform, there is again a drop in expression where the exon ends. In the correlation matrices, we see how the last exon is split into two diagonal blocks, one for the coding region and the other for the non-coding region.

In HRAS (Fig. 4.9, 4.15, 4.21), the first exon of the first promoter is completely recognized $\forall r$, we also observe a peak on the first exon of the third promoter (index ~ 250), however, it is relatively low, because isoforms whose promoter is upstream, decreasingly predict that exon as such; still regarding this exon, we see how the greater the r, the quicker is tends to predict it as intron. An interesting event is presented in HRAS around the index 700, our greedy generations have recognized an intronic region as exonic across most r experiments; such an area could be studied to find a possible novel exon. The second overall exon is well-defined for $r = 75$, and larger values extend it upstream, as it happens for the third exon, but more intensively at the latter. Two new insights in this gene are for nearby exons, $r = 200$ creates a bridge between exons three and four, and the small fifth exon tends to be disregarded with $50 \leq r \leq 150$, and the bridge effect brings it back to the plot, although with a smaller peak than the previously mentioned bridge. The last exon (from ~ 2750) or the last two exons, depending on the isoform, we notice again a steep drop in expression following the stop codon, also from the top generated isoforms, most end nearly after; however, there is a rise in expression corresponding to the last non-coding exon for $r \leq 75$. Note that the expression height is normalized by the time such a base is either an intron or an exon; therefore, the peaks in the last bases might be due to just a few of the generated isoforms classifying it as an exon, as the others consider that area as flanking DNA. The correlation matrices for $r \geq 150$ depict the high cross-correlation of the three exons in the middle, and with $r = 200$, there are intron blocks with high cross-correlation

that are up to 3000 nt away.

In MYC (Fig. 4.10, 4.16, 4.22), the first exon of the promoters before index 500 is very well defined $\forall r$. In this gene, the first intronic region has a noticeable amount of greedy generations that classify part of it as an exon, so a further study would be interesting. The second exon is also very well defined, especially downstream, but the corresponding 3'SS is visibly being recognized before the real one for $r \geq 150$. The last exon is not as certain as the others, but as we will see, the staircase of expression helps our binary classifier presented in the next section. With $r = 200$, the second and third exons are highly cross-correlated; note that they are not highly cross-correlated with the first exon because some greedy-generated isoforms end after the first exon.

KLF6 is a longer gene, with a high conservation of part of the first intronic region (Fig. 4.11, 4.17, 4.23). The first exon is well-defined $\forall r$, and the first intron, which has large high conservation regions with respect to the other genes seen so far, in this area, we notice a short peak, particularly with $r = 100$, corresponding to the first high conservation peak, and the rest of the intron is not as clearly defined. The second exon begins around 250 nt earlier $\forall r$, but has a clear drop on the correct 5'SS. The second intron of most known isoforms is also an exon in the first depicted isoform in the UCSC browser; our greedy generations tend to classify that region as an intron, with the exception of two small expression peaks visible with $r = 150$ and $r = 200$. The third exon is well-defined for $r \geq 150$, but somewhat recognized for smaller r as well. The 3'SS of the last exon is well-defined for $50 \leq r \leq 100$, as greater ranges r begin the exon earlier. Such an exon is almost fully non-coding, despite the high conservation in some areas; the exon expression or our greedy-generated isoforms are uncorrelated to coding or non-coding, or to the conservation. A further study would be needed to evaluate how LoRNA$^{\text{SH}}$ acts here. The top isoforms with $r = 75$ have a great variety of polyadenylation sites. For $r \geq 100$, there is an increasing number of cross-correlated intron blocks all over the extent of the gene.

In MDM2, as previously stated, we are acting on a completely non-coding, low-conserved region of around 1800 nt (Fig. 4.12, 4.18, 4.24), simply due to the single transcript corresponding to this gene being on this region inside the IIGM's dataset. We notice a very low variance across generations of a single r experiment, which is due to most sequences ending early.

In SRSF1 (Fig. 4.13, 4.19, 4.25), the first exon's 5'SS are not as good as most of the previous genes. The second exon is correctly classified for $r \geq 150$. The third and fourth exons of the second reference isoform from the top are well-defined for $r \geq 100$, although a bridge effect is noticed between them; however, in this case, the bridged region is also classifiable as an exon on other reference isoforms, e.g., the third from the top. After these exons, the reference isoforms are structured very differently, but most are within the non-coding region, despite the high conservation.

In this region, there are clear exon blocks present $\forall r$, i.e., indexes $\sim (3050 - 4200)$ and from index ~ 4600 on. We do notice a strong negative cross-correlation between the correct exon classifications and these exon blocks in the non-coding region that do not match the structure of known ones, probably because most generated isoforms have the polyadenylation site earlier, as observed in the top ones from $r = 75$; the negative correlation is particularly visible with $r = 200$.

4.3.2 Perplexity

We measured the perplexity of our greedy algorithm for each choice it had to make, as well as when a change between exon and intron was made, to evaluate the certainty or randomness of the greedy algorithm, as described in Sec. 3.4.2. In Fig. 4.26, depicting the perplexity of every decision point is clear that the algorithm is very certain of most decisions; this near 1 perplexity is due to the many evaluations it makes within a single state of intron or exon, and the monotonic decreasing perplexity with r is due to enlarging the gap between probabilities with more and more bases it analyses, and again considering the same state bias, it must decrease with r.

So to remove the same state bias, we plot the perplexity of the decisions made only when a change of state was observed (Fig. 4.27), the trend is not monotonic anymore and perplexity covers its full codomain, i.e., between 1 and 2. Interestingly, the median perplexity for $50 \leq r \leq 150$ is noticeably lower than the rest, even for $r = 200$, meaning that having a larger context could increase the uncertainty; as we observed in the previous sections, the larger r does not always mean a better result.

Analysing the perplexity for each gene (Fig. 4.28, 4.29), we observe the same decreasing median perplexity trend with r, however, when considering only the points in which it changes state, there is a variety of behaviors, so it is clear that the algorithm's perplexity of a certain r depends on the gene. FOXA1 has a $75 \leq r \leq 150$ below the rest, but $r = 100$ is the most certain one. MYC and HRAS have $50 \leq r \leq 150$ with fairly similar median perplexity. The minimum value for KLF6 is $r = 50$. MDM2 has the most diverse distributions due to the nature of the sequence it has analyzed, which is outside the coding region. And SRSF1 has the minimum perplexity at $r = 150$.

Together with our descriptive analysis of the exon expressions, we notice that the range $50 \leq r \leq 150$ contains not only the best results, but also with more certainty.

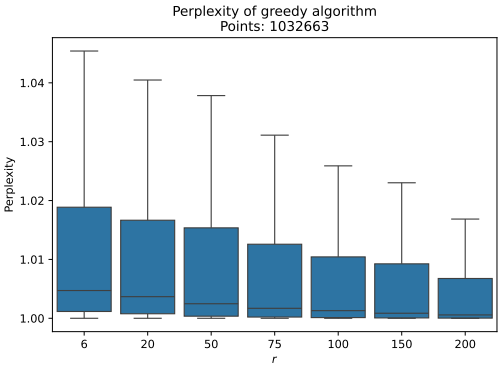

Figure 4.26: Perplexity of all decisions made by the Greedy algorithm

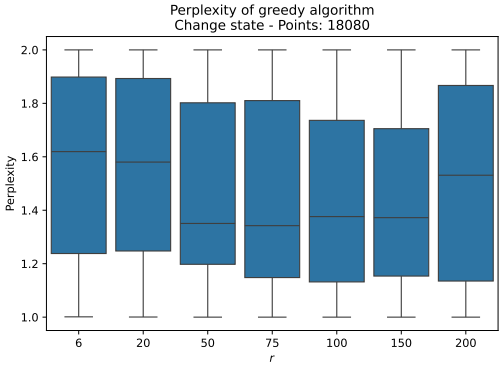

Figure 4.27: Perplexity of decisions that change state made by the Greedy algorithm

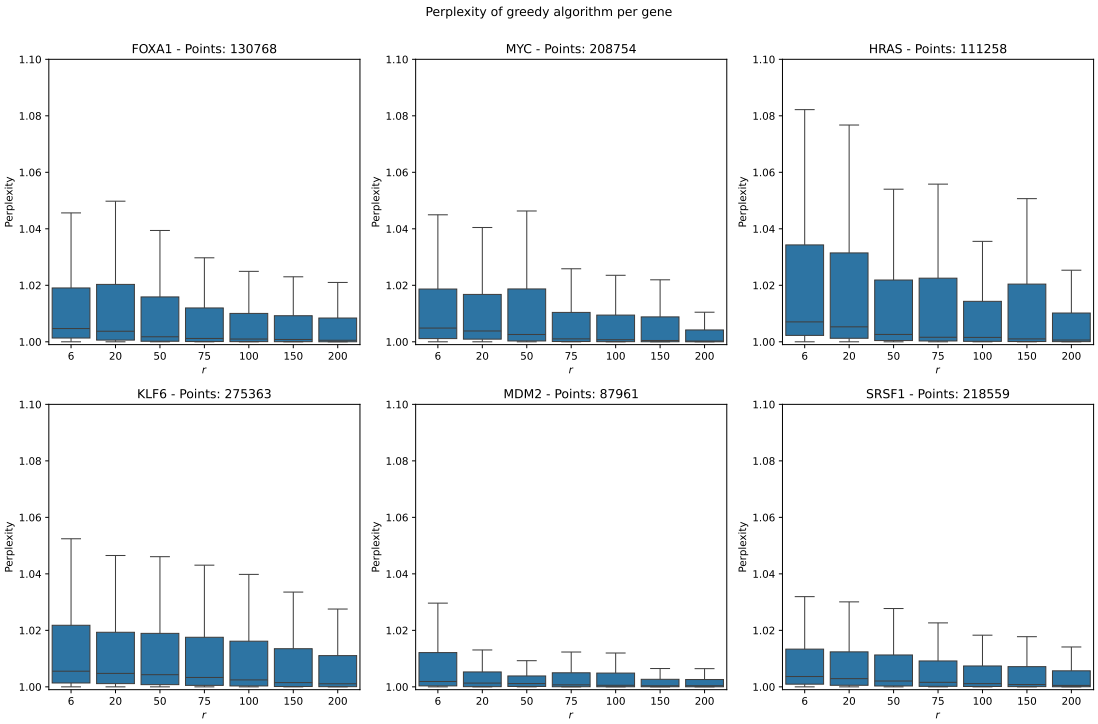

Figure 4.28: Perplexity of all decisions made by the Greedy algorithm by gene

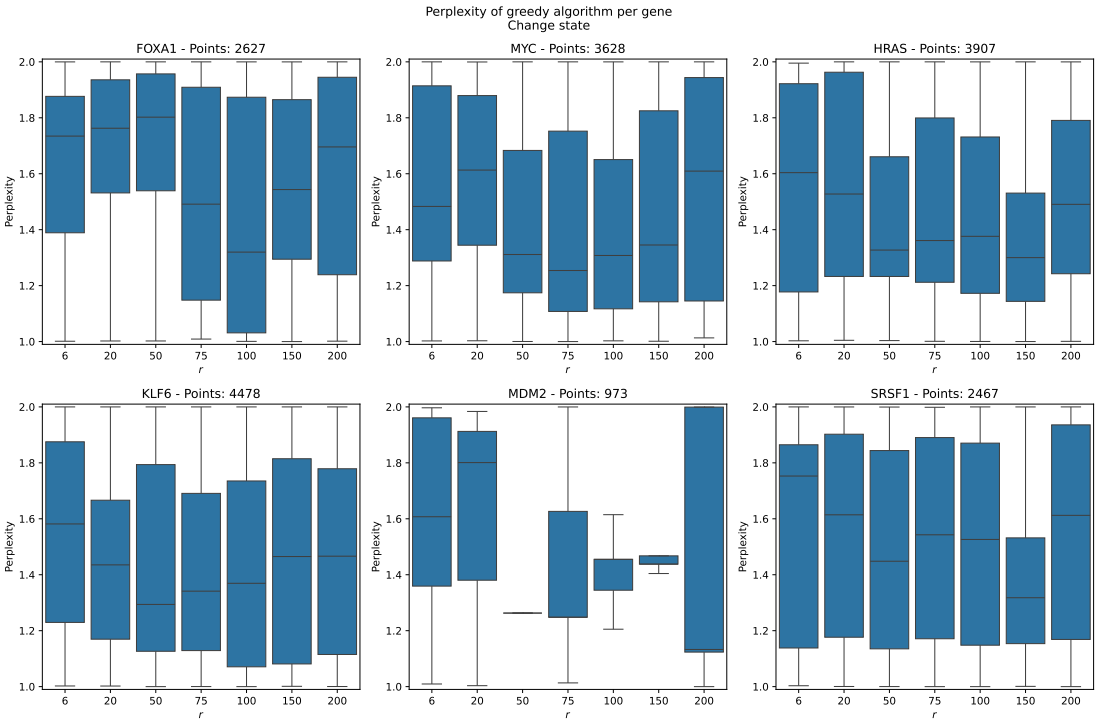

Figure 4.29: Perplexity of decisions that change state made by the Greedy algorithm by gene

4.4 Base-level Binary Classifier

We exploit the global information among all greedy generated isoforms of the same promoter site by creating a single-nucleotide-level exon classifier. Given a variable threshold t, we label a base x_i as an exon if its exon expression is above the threshold. The methods used to evaluate the classifier are both the AUC and the Long-expression-weighted AUC, as referred in Sec. 3.5.1.

4.4.1 ROC curves and AUC

In the figures 4.30-4.34, we plot the ROC curve of a single hand-picked isoform for each gene as an example; a more general evaluation including the 19 isoforms will follow. The five characters code in parentheses in the figures are the last part of a unique sequencing identifier from the dataset. In these examples, we notice just a few points below the random guesser classifier line, in FOXA1. In the HRAS example, most values of r have a similar curve with high AUCs, most with AUC $> 90\%$, while in KLF6, just a few r exceed 80% AUC. The MYC example

contains a curve with AUC = 1.000 with $r = 100$, and overall high values in the other ranges too. And finally, the example of SRSF1 contains a ROC curve for $r = 200$ that clearly struggles more than the rest, with AUC = 73%, and 14% apart from the second lowest AUC.

Note that the gene MDM2 was excluded from this evaluation because the only isoform available in the dataset is monoexonic, and the integral for the AUC is indefinite when the ground truth has only positive values. The *e15ae* FOXA1 isoform was excluded for being monoexonic too.

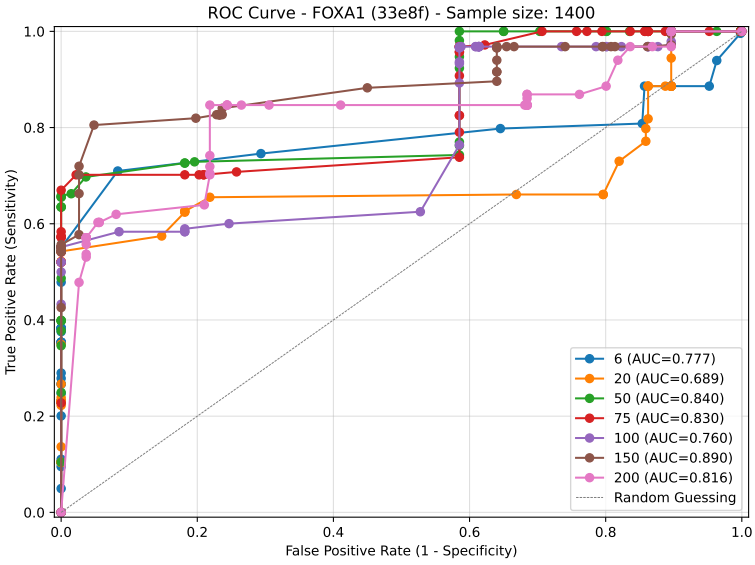

Figure 4.30: ROC curve example - FOXA1

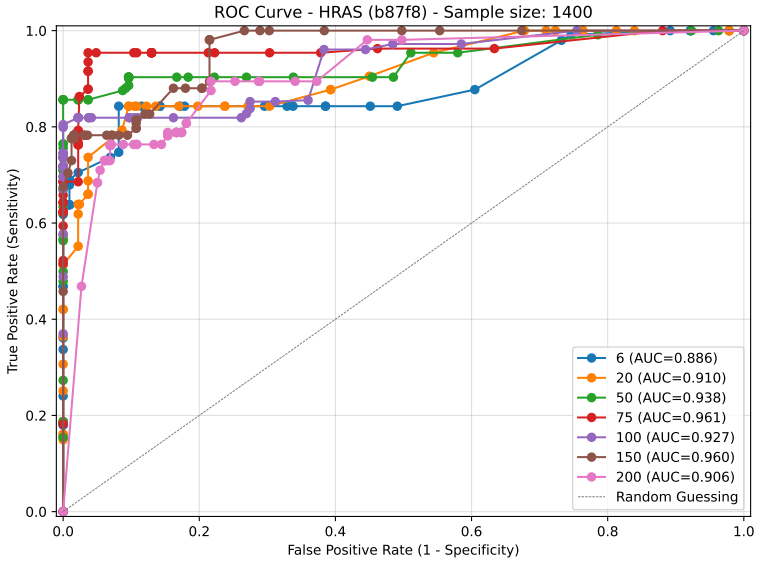

Figure 4.31: ROC curve example - HRAS

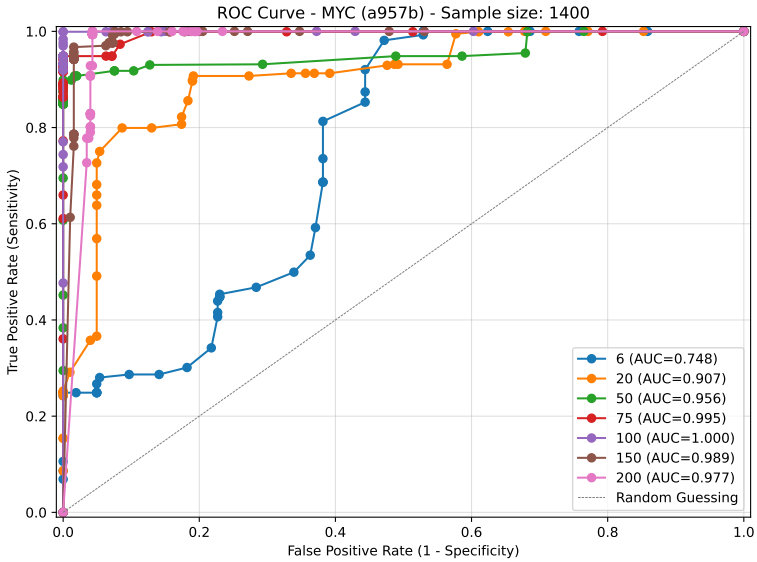

Figure 4.32: ROC curve example - MYC

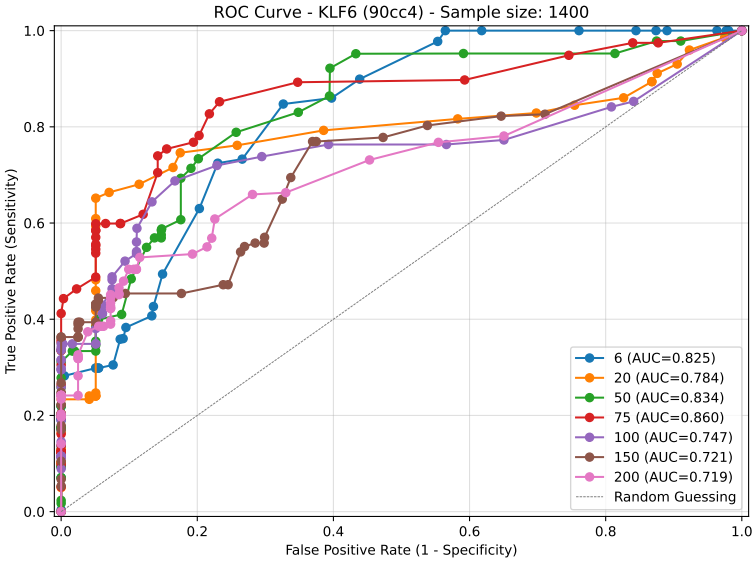

Figure 4.33: ROC curve example - KLF6

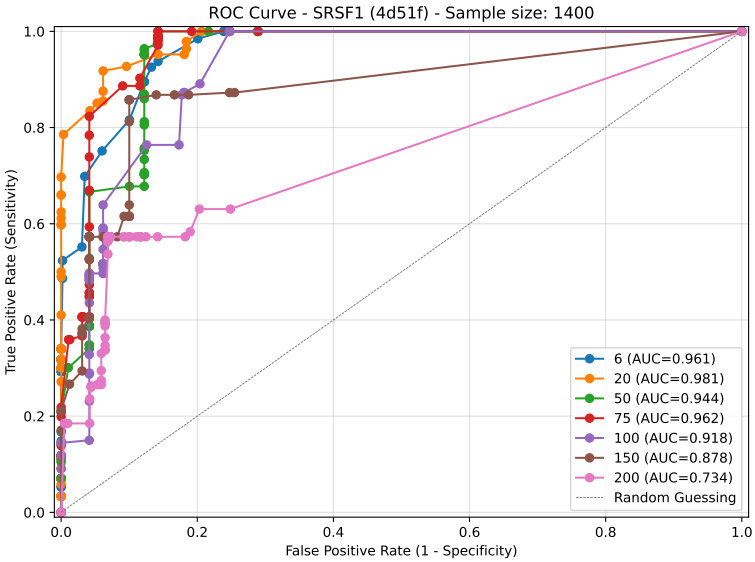

Figure 4.34: ROC curve example - SRSF1

Gene \ r	6	20	50	75	100	150	200	Gene mean
FOXA1	.847	.788	.890	.883	.834	**.923**	.764	.85
HRAS	.904	.904	.921	**.937**	.919	.931	.874	.91
MYC	.750	.909	.956	.993	**.995**	.985	.973	.94
KLF6	**.910**	.879	.882	.883	.799	.773	.764	.84
SRSF1	.901	.902	.885	**.908**	.905	.885	.853	.89
r mean	.86	.88	.91	**.92**	.89	.90	.85	

Table 4.5: Average AUC across 19 isoforms for multiple genes and ranges

Gene \ r	6	20	50	75	100	150	200	Gene mean
FOXA1	.767	.678	.832	.820	.747	**.883**	.803	.79
HRAS	.884	.891	.913	.942	.917	**.954**	.887	.91
MYC	.750	.909	.956	.993	**.995**	.985	.973	.94
KLF6	.822	.783	.838	**.861**	.740	.712	.708	.78
SRSF1	.950	**.957**	.931	.946	.920	.893	.799	.91
r mean	.83	.84	.89	**.91**	.86	.89	.83	

Table 4.6: Long-expression weighted average AUC across 19 isoforms for multiple genes and ranges

The AUC values averaged across isoforms for each pair of range r and gene are presented in Tab. 4.5, and the Long-read expression weighted average is presented in Tab. 4.6. To perform the final test, we have selected the range $r = 75$, as it has the best performance in both metrics across the five studied genes.

4.4.2 Final test

Having chosen $r = 75$, with 51 genes such that each has at least 3 non-monoexonic isoforms and a gene length below 10 knt, we have generated 500 isoforms with the greedy algorithm from each of the 100 known promoter sites from the reference, being related to 271 isoforms, and finally calculated the AUCs of the binary classifier, by gene weighting the isoforms' AUC with the long-read expression (Fig. 4.35), and by isoform, with or without seperating the known and novel ones (Fig. 4.37 and Fig. 4.36).

The known isoforms' AUC present a bimodal distribution with one peak near 100% and the other just under 90%, a phenomenon that was not observed among the novel isoforms, whose distribution has a tail towards lower values of AUC, having an outlier at 48.5%. Although the distributions are different, the quartiles, median, and average remain within a distance of less than 5%.

Furthermore, we studied the correlation between the AUC of the isoforms and the natural logarithm of their long-read expression in Fig. 4.38. There is a small correlation (PCC = 0.119) and a p-value of 5.668%, so the correlation is not conclusive.

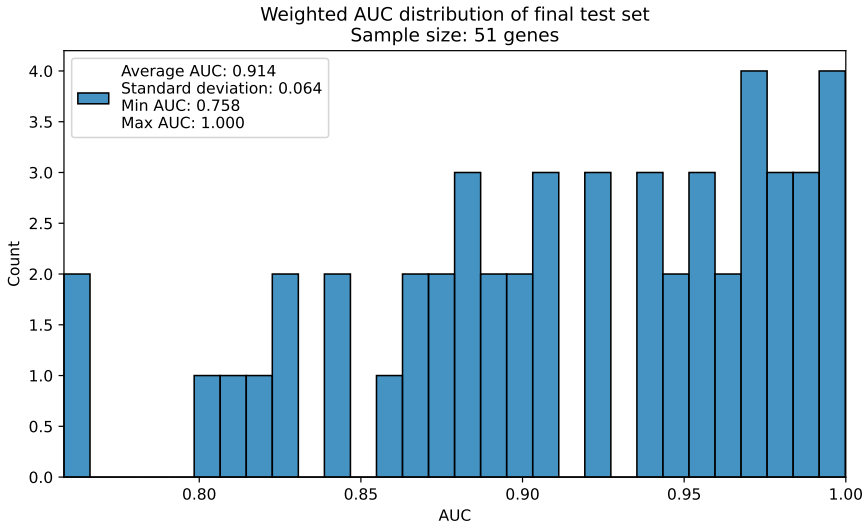

Figure 4.35: WAUC distribution on final test by genes.

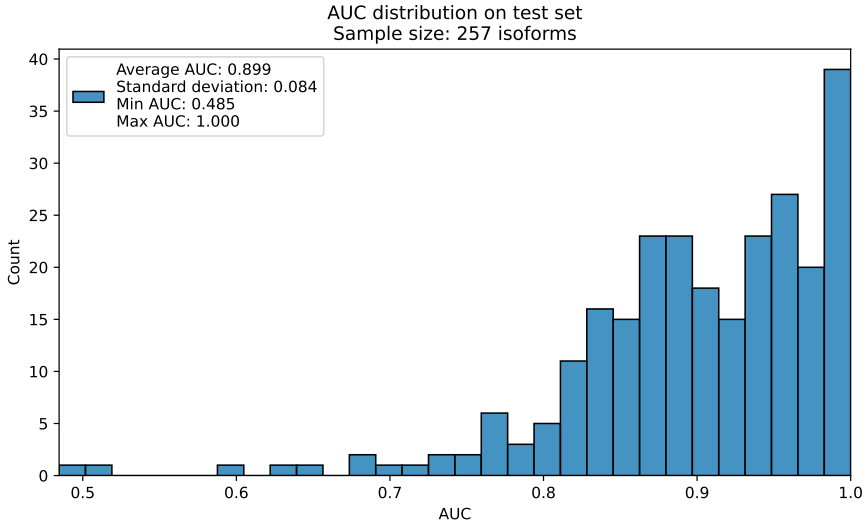

Figure 4.36: AUC distribution on final test by isoform.

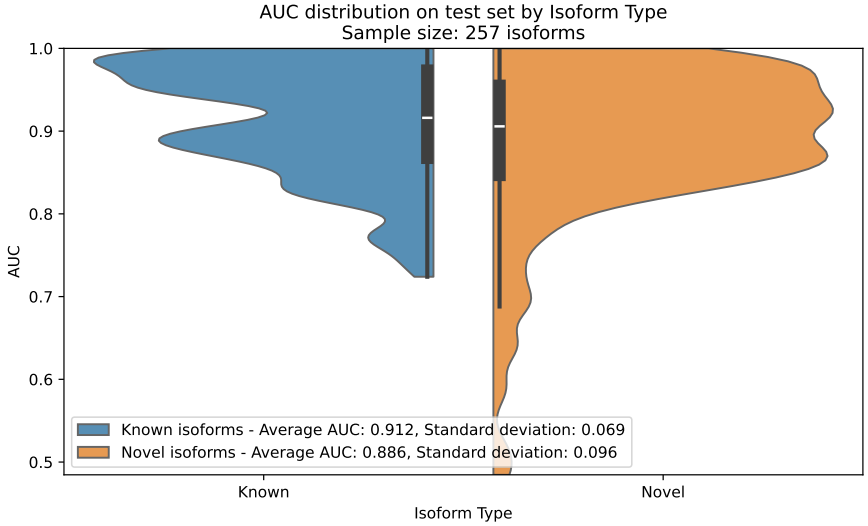

Figure 4.37: AUC distribution on final test by isoform type. Sample sizes by type: 136 known, and 121 novel.

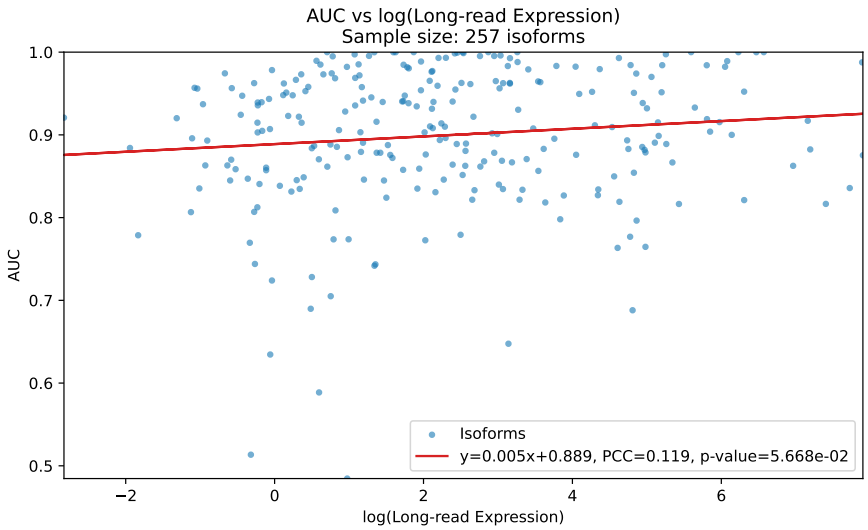

Figure 4.38: AUC vs log(Long-read expression)

4.5 Hardware resources in Greedy algorithm

As described in Sec. 3.4.5, we analyze the time and VRAM complexity with the computational resources provided by HPC@PoliTO [100], using an Intel Xeon Gold CPU and an NVIDIA A40 GPU.

4.5.1 Algorithm time complexity

We measured the execution, GPU, and CPU times, out of 25000 generations of variable lengths. From Fig. 4.39 we observe that the quadratic relationship between length L and time t is a good fit for $L < 10$ knt; there is no significant reduction of error by using a third-degree polynomial fit. Furthermore, we notice that some isoforms take noticeably more time to be generated, to inspect this phenomenum, we have taken the generated isoform that took the longest for each gene, and we observed that for execution time and GPU times, 51/51 were associated with the first isoform generated for a given promoter, we can infer that there is some optimization occurring that takes advantage of the first computation.

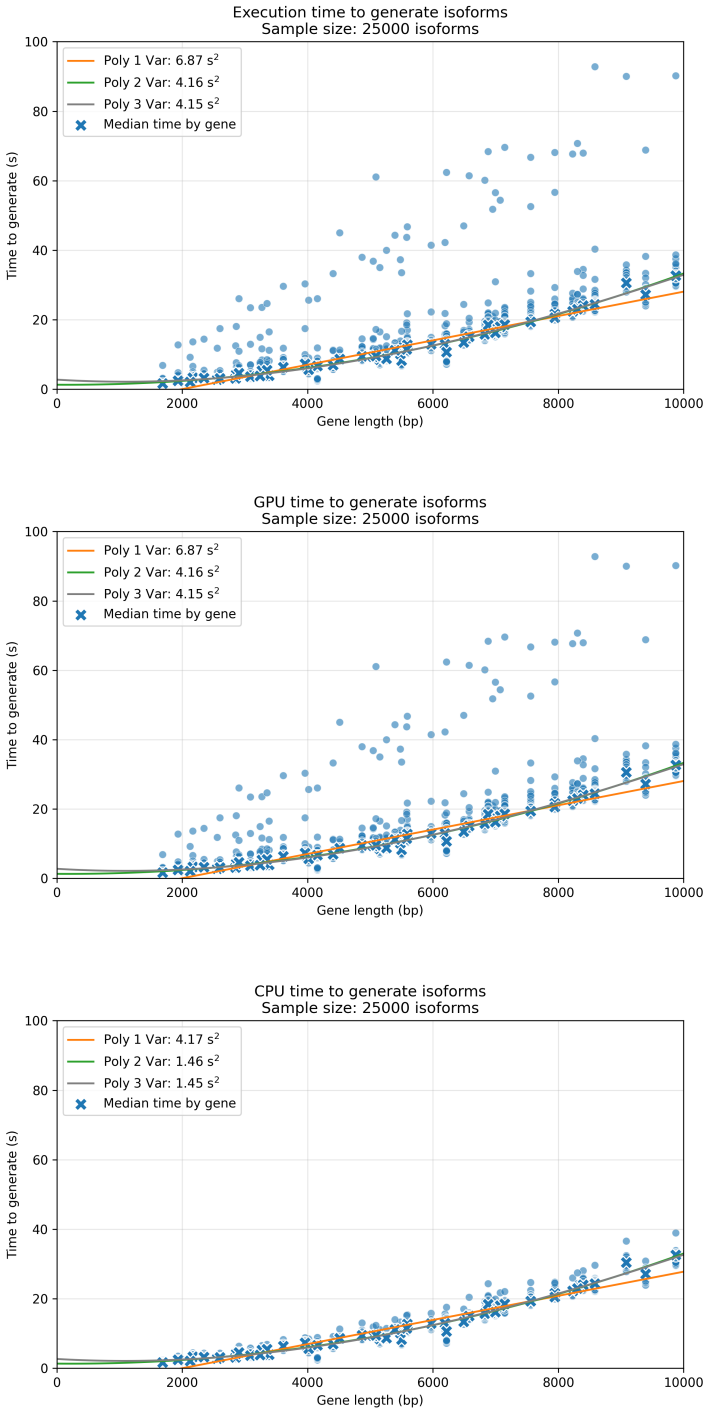

Figure 4.39: Execution, GPU, and CPU times in Greedy algorithm

4.5.2 GPU memory usage

The computational limitations are related to the VRAM usage in the GPU, noticing that the model occupies by itself only 6MB (Sec 2.3.5), the major memory allocation is related to feed-forwarding the sequence; the longer the sequence, the more it allocates. The compared configurations of GPUs are presented in Tab. 4.7. These results were obtained by iteratively selecting increasing-length genes within the dataset, tokenizing them, and computing the probability tensor with LoRNA^{SH}, until a memory error occurred. With multiple GPUs, one can assign different layers to each unit. The winning strategy here is to assign the most memory-hungry layers uniformly distributed among the machines, which, in our case, are the Attention layers.

Figure 4.40 presents the distribution of gene lengths present on IIGM's dataset (Sec. 3.1.1), together with the GPU limits.

The model has the following structure, in order: 4× Hyena Blocks, 1× Attention, 3× Hyena Blocks, 1× Attention, 3× Hyena Blocks. When using 4 GPUs, we assigned the first four layers of Hyena blocks to one GPU and one Attention layer, along with three Hyena blocks, to the other GPUs, in a way that minimizes memory swaps.

Notice that the context of the model is of 65536 nt, so in our results, we managed to include the whole context with a single RTX A40 GPU. The other two important maximum lengths in this thesis are the 10 knt, used in our algorithm (Sec. 3.4), and 20 knt, used in the pure NTP generations (Sec. 3.3). More nucleotides beyond the maximum context are only stored in the GPU memory to be output when computations are done; no more than 65536 bases are included in the computations. No tests were done to evaluate the model metrics on longer contexts, which would exclude special tokens from the beginning of the sequence, i.e., [CLS], H, and S.

GPUs	VRAM (GB)	Max nucleotides (knt)	Genome coverage (%)
1× RTX A5000	24	30	63.38
1× RTX A40	48	85	86.45
4× RTX A40	192	284	98.28

Table 4.7: GPU configurations

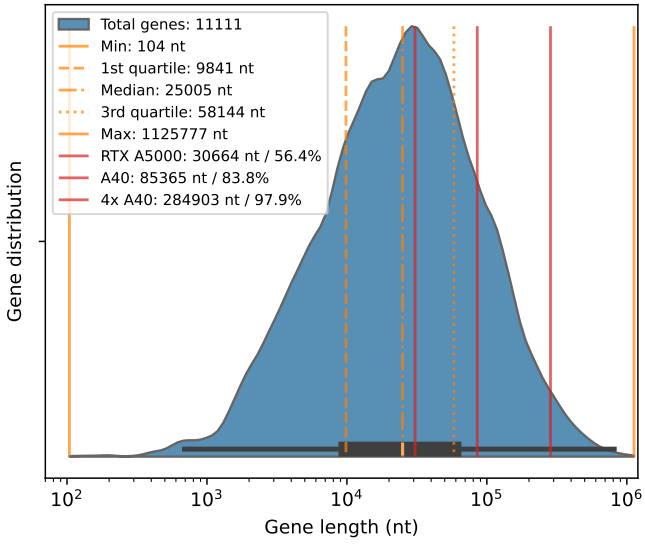

Figure 4.40: GPU limits on IIGM's dataset using LoRNA$^{\text{SH}}$

A more specific study measuring the peak VRAM was made for $L < 10$ knt, with 25000 generations, and we observe there is a perfect linear dependency in that region (Fig. 4.41).

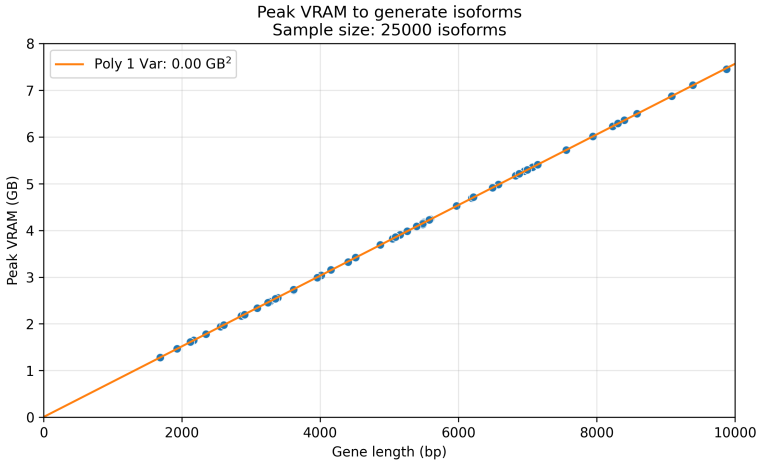

Figure 4.41: Peak VRAM usage during Greedy algorithm

Chapter 5

Conclusion

The problem of discovering new isoforms is important for the advancement of knowledge about many diseases, including cancer, and it is useful even in clinical applications. Sequencing all mRNA isoforms from a replicate is mathematically improbable with current technology, especially for isoforms that are lowly expressed. Therefore, a computational approach to predicting possible isoforms is an interesting problem. Even if not observed, one can study their likely behavior. In this thesis, we have explored a naive way of creating possible isoforms of a gene combinatorially, but it has proved to be both inaccurate and unfeasible, so we have employed LoRNA$^{\text{SH}}$, an LLM pretrained on 26 tumoral cell lines with mRNA sequences, and extracted its NTP for making reasonable decisions in our greedy algorithm.

In our experiments with the genes FOXA1, HRAS, MYC, KLF6, MDM2, and SRSF1, a discrete number of context expander ranges were explored, known as the hyperparameter r. First, we perform a qualitative analysis on the exon expression of a cluster of greedy-generated isoforms compared to the reference GRCh37, together with the perplexity study for each r when the algorithm decides to change state. We notice how small ranges ($6 \leq r \leq 20$) usually do not help our algorithm decide whether to continue the isoform as an intron or an exon; we see it both from the noisier patterns in the exon expression and from the higher perplexity. Considering all genes in the range $r \in (50, 150)$ had lower perplexity when a state change is observed, it is interesting to notice how the perplexity rises again with $r = 200$; but different genes had different distributions. In general, we have observed that the first exon is usually the easiest to be correctly recognized, and that exon portions outside the coding region are harder to be predicted as such. We notice in some cases exon expression peaks in our generations that do not match the reference's exons, meaning it could have found a novel exon. Increasing r, sometimes we notice a premature exon start or the bridging effect, uniting nearby exons; also, although it is not always the case, exons shorter than the r value might be skipped.

In order to take the most advantage from a global perspective of all generated

isoforms, we created a base-level exon classifier using a threshold on the generated expression. This final algorithm is evaluated with the AUC. The best performing range hyperparameter in the first six genes cited above was $r = 75$ in both average AUC across isoforms and genes, and with the long-read expression weighted average. So, use used this value of r to make the final test with 51 genes, obtaining AUC $= 89.9\% \pm 8.4\%$ (mean $\pm$ standard deviation) across all 257 isoforms present in the long-read IIGM dataset of PC3 cells, from which 136 are known with $\text{AUC}_{\text{known}} = 91.2\% \pm 6.9\%$, and 121 are novel with $\text{AUC}_{\text{novel}} = 88.6\% \pm 9.6\%$. There is a bias towards known isoforms, but the prediction of novel isoforms is still remarkable.

This study also includes the reproduction of some of the results presented in the LoRNA$^{\text{SH}}$ paper, such as the distribution of lengths and counts of introns and exons, sequence lengths, and SpliceAI probability of each used splicing site of the pure generations, i.e., isoforms generated just with the prompt HS. Our results did not match the paper's, which might be due to different processing strategies, but some similarities when comparing to both of our references are noticed. Lastly, we present the computational resources used for the greedy algorithm in terms of computational time and VRAM, varying the gene length.

5.1 Future works

This work presented the first steps towards exploiting NTP from LLMs to produce probable mRNA isoforms, but much can still be studied and the algorithm improved. For example, longer genes could be tested, as well as testing the performance with different species, and fine-tuning LoRNA$^{\text{SH}}$, which was trained only with the longest isoforms, so training with shorter and novel isoforms from other datasets might be beneficial. Another improvement is studying the choice of r for each gene based on its known structure, such as length, number of exons, average exon and intron lengths, and their conservation. Our algorithm has some biases that enforce some biological features, such as promoter site, and most importantly, we have used only U2 intron splicing signals, which are used over 99% of eukaryotic introns, but there are other splicing options, such as the U12. Studying possibilities for predicting the relative expression across isoforms would be interesting. All of it, keeping up to date with the most recent versions of the model. And finally, a biological study of our produced novel isoforms, such as verifying they are protein-coding, the 3D structure of the protein, and a prediction of how this protein would interact in biological pathways.

Bibliography

[1] Freddie Bray, Mathieu Laversanne, Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Isabelle Soerjomataram, and Ahmedin Jemal. «Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries». In: *CA: A Cancer Journal for Clinicians* 74.3 (2024), pp. 229–263. DOI: https://doi.org/10.3322/caac.21834. eprint: https://acsjournals.onlinelibrary.wiley.com/doi/pdf/10.3322/caac.21834. URL: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21834 (cit. on p. 1).

[2] Yuanjiao Zhang, Jinjun Qian, Chunyan Gu, and Ye Yang. «Alternative splicing and cancer: a systematic review». In: *Signal Transduction and Targeted Therapy* 6.1 (Feb. 2021), p. 78 (cit. on pp. 1, 11).

[3] Huiping Chen, Jingqun Tang, and Juanjuan Xiang. «Alternative Splicing in Tumorigenesis and Cancer Therapy». en. In: *Biomolecules* 15.6 (May 2025) (cit. on pp. 1, 12, 13).

[4] Robert K Bradley and Olga Anczukow. «RNA splicing dysregulation and the hallmarks of cancer». In: *Nature Reviews Cancer* 23.3 (Mar. 2023), pp. 135–155 (cit. on pp. 1, 11).

[5] Deanna M Church et al. «Modernizing reference genome assemblies». en. In: *PLoS Biol* 9.7 (July 2011), e1001091 (cit. on pp. 2, 21).

[6] Adam Siepel, Katherine S. Pollard, and David Haussler. «New Methods for Detecting Lineage-Specific Selection». In: *Research in Computational Molecular Biology*. Ed. by Alberto Apostolico, Concettina Guerra, Sorin Istrail, Pavel A. Pevzner, and Michael Waterman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 190–205. ISBN: 978-3-540-33296-1 (cit. on pp. 2, 23).

[7] Johann Friedrich II Miescher. «Die Spermatozoen einiger Wirbelthiere. Ein Beitrag zur Histochemie». In: *Verhandlungender Naturforschenden Gesellschaft in Basel 6: 138–208* (1874) (cit. on p. 4).

[8] Andrew Gates and Richard Bowater. «Nucleotides: Structure and Properties». In: *Encyclopedia of Life Sciences* (Feb. 2015). DOI: 10.1002/9780470015902.a0001333.pub3 (cit. on p. 4).

[9] International Human Genome Sequencing Consortium. «Finishing the euchromatic sequence of the human genome». In: *Nature* 431.7011 (Oct. 2004), pp. 931–945. ISSN: 1476-4687. DOI: 10.1038/nature03001. URL: https://doi.org/10.1038/nature03001 (cit. on p. 4).

[10] CD Genomics Blog. *Sequencing Read Length: Everything You Need to Know | CD Genomics Blog*. 2024. URL: https://www.cd-genomics.com/blog/sequencing-read-length-comprehensive/ (visited on 10/07/2025) (cit. on p. 5).

[11] F Sanger, S Nicklen, and A R Coulson. «DNA sequencing with chain-terminating inhibitors». en. In: *Proc Natl Acad Sci U S A* 74.12 (Dec. 1977), pp. 5463–5467 (cit. on p. 5).

[12] Edward Chait, Guy Page, and Michael Hunkapiller. «Battle of the DNA sequencers». In: *Nature* 333.6172 (June 1988), pp. 477–478. ISSN: 1476-4687. DOI: 10.1038/333477a0. URL: https://doi.org/10.1038/333477a0 (cit. on p. 5).

[13] Inc. Illumina. *NextSeq 1000 & 2000 System Specifications | Output, run time, & more*. 2025. URL: https://www.illumina.com/systems/sequencing-platforms/nextseq-1000-2000/specifications.html (visited on 09/18/2025) (cit. on p. 5).

[14] Barton E Slatko, Andrew F Gardner, and Frederick M Ausubel. «Overview of Next-Generation Sequencing Technologies». en. In: *Curr Protoc Mol Biol* 122.1 (Apr. 2018), e59 (cit. on p. 5).

[15] Inc. Illumina. *An introduction to Next-Generation Sequencing Technology*. 2017. URL: https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf (visited on 08/09/2025) (cit. on p. 5).

[16] Inc. Illumina. *Phasing Correction*. 2023. URL: https://support-docs.illumina.com/IN/NextSeq_550-500/Content/IN/PhasingCorrection_RTA2.htm (visited on 08/09/2025) (cit. on p. 5).

[17] Yunhao Wang, Yue Zhao, Audrey Bollas, Yuru Wang, and Kin Fai Au. «Nanopore sequencing technology, bioinformatics and applications». In: *Nature Biotechnology* 39.11 (Nov. 2021), pp. 1348–1365. ISSN: 1546-1696. DOI: 10.1038/s41587-021-01108-x. URL: https://doi.org/10.1038/s41587-021-01108-x (cit. on p. 6).

[18] Oxford Nanopore Technologies plc. *Dorado Documentation*. 2025. URL: https://software-docs.nanoporetech.com/dorado/latest/ (visited on 08/13/2025) (cit. on p. 6).

[19] Jasper Verwilt, Pieter Mestdagh, and Jo Vandesompele. «Artifacts and biases of the reverse transcription reaction in RNA sequencing». en. In: *RNA* 29.7 (Mar. 2023), pp. 889–897 (cit. on p. 6).

[20] Clifford A Meyer and X Shirley Liu. «Identifying and mitigating bias in next-generation sequencing methods for chromatin biology». en. In: *Nat Rev Genet* 15.11 (Sept. 2014), pp. 709–721 (cit. on p. 6).

[21] Xanthi-Lida Katopodi, Oguzhan Begik, and Eva Maria Novoa. «Toward the use of nanopore RNA sequencing technologies in the clinic: challenges and opportunities». In: *Nucleic Acids Research* 53.5 (Mar. 2025), gkaf128. ISSN: 1362-4962. DOI: 10.1093/nar/gkaf128. eprint: https://academic.oup.com/nar/article-pdf/53/5/gkaf128/62341879/gkaf128.pdf. URL: https://doi.org/10.1093/nar/gkaf128 (cit. on pp. 6, 7).

[22] National Human Genome Research Institute. *Talking Glossary of Genetic Terms | NHGRI*. 2025. URL: https://www.genome.gov/genetics-glossary (visited on 08/09/2025) (cit. on p. 6).

[23] Robert Carter and Guy Drouin. «Structural differentiation of the three eukaryotic RNA polymerases». In: *Genomics* 94.6 (2009), pp. 388–396. ISSN: 0888-7543. DOI: https://doi.org/10.1016/j.ygeno.2009.08.011. URL: https://www.sciencedirect.com/science/article/pii/S0888754309002043 (cit. on p. 7).

[24] Eric Wang and Iannis Aifantis. «RNA Splicing and Cancer». In: *Trends in Cancer* 6.8 (2020), pp. 631–644. ISSN: 2405-8033. DOI: https://doi.org/10.1016/j.trecan.2020.04.011. URL: https://www.sciencedirect.com/science/article/pii/S2405803320301412 (cit. on p. 7).

[25] Wikipedia Commons. *File:DNA alternative splicing.gif*. 2014. URL: https://commons.wikimedia.org/wiki/File:DNA_alternative_splicing.gif (visited on 10/08/2025) (cit. on p. 8).

[26] Laura Poliseno, Martina Lanza, and Pier Paolo Pandolfi. «Coding, or non-coding, that is the question». In: *Cell Research* 34.9 (Sept. 2024), pp. 609–629. ISSN: 1748-7838. DOI: 10.1038/s41422-024-00975-8. URL: https://doi.org/10.1038/s41422-024-00975-8 (cit. on p. 8).

[27] Matthew J. Betts, Roderic Guigó, Pankaj Agarwal, and Robert B. Russell. «Exon structure conservation despite low sequence similarity: a relic of dramatic events in evolution?» In: *The EMBO Journal* 20.19 (2001), pp. 5354–5360. DOI: https://doi.org/10.1093/emboj/20.19.5354. eprint: https://www.embopress.org/doi/pdf/10.1093/emboj/20.19.5354. URL: https://www.embopress.org/doi/abs/10.1093/emboj/20.19.5354 (cit. on p. 8).

[28] John S Mattick. «Non-coding RNAs: the architects of eukaryotic complexity». In: *EMBO reports* 2.11 (2001), pp. 986–991. DOI: https://doi.org/10.1093/embo-reports/kve230. eprint: https://www.embopress.org/doi/pdf/10.1093/embo-reports/kve230. URL: https://www.embopress.org/doi/abs/10.1093/embo-reports/kve230 (cit. on p. 8).

[29] Barbara Ruskin and Michael R. Green. «Role of the 3' splice site consensus sequence in mammalian pre-mRNA splicing». In: *Nature* 317.6039 (Oct. 1985), pp. 732–734. ISSN: 1476-4687. DOI: 10.1038/317732a0. URL: https://doi.org/10.1038/317732a0 (cit. on p. 8).

[30] Schraga H Schwartz, João Silva, David Burstein, Tal Pupko, Eduardo Eyras, and Gil Ast. «Large-scale comparative analysis of splicing signals and their corresponding splicing factors in eukaryotes». en. In: *Genome Res* 18.1 (Nov. 2007), pp. 88–103 (cit. on p. 8).

[31] Josep F Abril, Robert Castelo, and Roderic Guigó. «Comparison of splice sites in mammals and chicken». en. In: *Genome Res* 15.1 (Dec. 2004), pp. 111–119 (cit. on pp. 8, 24).

[32] Cindy L Will and Reinhard Lührmann. «Spliceosome structure and function». en. In: *Cold Spring Harb Perspect Biol* 3.7 (July 2011) (cit. on p. 8).

[33] G W Beadle and E L Tatum. «Genetic Control of Biochemical Reactions in Neurospora». en. In: *Proc Natl Acad Sci U S A* 27.11 (Nov. 1941), pp. 499–506 (cit. on p. 9).

[34] NobelPrize.org. *The Nobel Prize in Physiology or Medicine 1958*. 2025. URL: https://www.nobelprize.org/prizes/medicine/1958/summary/ (visited on 08/11/2025) (cit. on p. 9).

[35] Louise T. Chow, Richard E. Gelinas, Thomas R. Broker, and Richard J. Roberts. «An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA». In: *Cell* 12.1 (Sept. 1977), pp. 1–8. ISSN: 0092-8674. DOI: 10.1016/0092-8674(77)90180-5. URL: https://doi.org/10.1016/0092-8674(77)90180-5 (cit. on p. 9).

[36] Susan M. Berget, Claire Moore, and Phillip A. Sharp. «Spliced segments at the 5'; terminus of adenovirus 2 late mRNA*». In: *Proceedings of the National Academy of Sciences* 74.8 (1977), pp. 3171–3175. DOI: 10.1073/pnas.74.8.3171. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.74.8.3171. URL: https://www.pnas.org/doi/abs/10.1073/pnas.74.8.3171 (cit. on p. 9).

[37] Eric T. Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F. Kingsmore, Gary P. Schroth, and Christopher B. Burge. «Alternative isoform regulation in human tissue transcriptomes». In: *Nature* 456.7221 (Nov. 2008), pp. 470–476. ISSN: 1476-4687. DOI: 10.1038/nature07509. URL: https://doi.org/10.1038/nature07509 (cit. on p. 10).

[38] Qun Pan, Ofer Shai, Leo J. Lee, Brendan J. Frey, and Benjamin J. Blencowe. «Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing». In: *Nature Genetics* 40.12 (Dec. 2008), pp. 1413–1415. ISSN: 1546-1718. DOI: 10.1038/ng.259. URL: https://doi.org/10.1038/ng.259 (cit. on p. 10).

[39] Min-Sik Kim et al. «A draft map of the human proteome». en. In: *Nature* 509.7502 (May 2014), pp. 575–581 (cit. on p. 10).

[40] Francisco E. Baralle and Jimena Giudice. «Alternative splicing as a regulator of development and tissue identity». In: *Nature Reviews Molecular Cell Biology* 18.7 (July 2017), pp. 437–451. ISSN: 1471-0080. DOI: 10.1038/nrm.2017.27. URL: https://doi.org/10.1038/nrm.2017.27 (cit. on p. 10).

[41] David Nikom and Sika Zheng. «Alternative splicing in neurodegenerative disease and the promise of RNA therapies». In: *Nature Reviews Neuroscience* 24.8 (Oct. 2023), pp. 457–473. ISSN: 1471-0048. DOI: 10.1038/s41583-023-00717-6. URL: https://doi.org/10.1038/s41583-023-00717-6 (cit. on p. 10).

[42] Pingping Ren, Luying Lu, Shasha Cai, Jianghua Chen, Weiqiang Lin, and Fei Han. «Alternative Splicing: A New Cause and Potential Therapeutic Target in Autoimmune Disease». In: *Frontiers in Immunology* Volume 12 - 2021 (2021). ISSN: 1664-3224. DOI: 10.3389/fimmu.2021.713540. URL: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2021.713540 (cit. on p. 10).

[43] Jen-Yang Tang, Jin-Ching Lee, Ming-Feng Hou, Chun-Lin Wang, Chien-Chi Chen, Hurng-Wern Huang, and Hsueh-Wei Chang. «Alternative splicing for diseases, cancers, drugs, and databases». en. In: *ScientificWorldJournal* 2013 (May 2013), p. 703568 (cit. on pp. 10, 11).

[44] Mariano A. Garcia-Blanco, Andrew P. Baraniak, and Erika L. Lasda. «Alternative splicing in disease and therapy». In: *Nature Biotechnology* 22.5 (May 2004), pp. 535–546. ISSN: 1546-1696. DOI: 10.1038/nbt964. URL: https://doi.org/10.1038/nbt964 (cit. on p. 10).

[45] Yan Wang et al. «Mechanism of alternative splicing and its regulation». en. In: *Biomed Rep* 3.2 (Dec. 2014), pp. 152–158 (cit. on p. 10).

[46] Francisco Gimeno-Valiente, Gerardo López-Rodas, Josefa Castillo, and Luis Franco. «The Many Roads from Alternative Splicing to Cancer: Molecular Mechanisms Involving Driver Genes». en. In: *Cancers (Basel)* 16.11 (June 2024) (cit. on p. 10).

[47] Douglas L Black. «Mechanisms of alternative pre-messenger RNA splicing». en. In: *Annu Rev Biochem* 72 (Feb. 2003), pp. 291–336 (cit. on p. 10).

[48] Qian Zhang, Yuxi Ai, and Omar Abdel-Wahab. «Molecular impact of mutations in RNA splicing factors in cancer». In: *Molecular Cell* 84.19 (Oct. 2024), pp. 3667–3680. ISSN: 1097-2765. DOI: 10.1016/j.molcel.2024.07.019. URL: https://doi.org/10.1016/j.molcel.2024.07.019 (cit. on p. 11).

[49] Shipra Das, Olga Anczuków, Martin Akerman, and Adrian R. Krainer. «Oncogenic Splicing Factor SRSF1 Is a Critical Transcriptional Target of MYC». In: *Cell Reports* 1.2 (Feb. 2012), pp. 110–117. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2011.12.001. URL: https://doi.org/10.1016/j.celrep.2011.12.001 (cit. on p. 11).

[50] Koh Miura, Wataru Fujibuchi, and Iwao Sasaki. «Alternative pre-mRNA splicing in digestive tract malignancy». en. In: *Cancer Sci* 102.2 (Dec. 2010), pp. 309–316 (cit. on p. 11).

[51] Gregory M Hayes, Patricia E Carrigan, Alison M Beck, and Laurence J Miller. «Targeting the RNA splicing machinery as a novel treatment strategy for pancreatic carcinoma». en. In: *Cancer Res* 66.7 (Apr. 2006), pp. 3819–3827 (cit. on p. 12).

[52] Gregory M Hayes, Patricia E Carrigan, and Laurence J Miller. «Serine-arginine protein kinase 1 overexpression is associated with tumorigenic imbalance in mitogen-activated protein kinase pathways in breast, colonic, and pancreatic carcinomas». en. In: *Cancer Res* 67.5 (Mar. 2007), pp. 2072–2080 (cit. on p. 12).

[53] Zhidong Wang and Yu J Cao. «Adoptive Cell Therapy Targeting Neoantigens: A Frontier for Cancer Research». en. In: *Front Immunol* 11 (Mar. 2020), p. 176 (cit. on p. 13).

[54] Inés Zugasti et al. «CAR-T cell therapy for cancer: current challenges and future directions». In: *Signal Transduction and Targeted Therapy* 10.1 (July 2025), p. 210. ISSN: 2059-3635. DOI: 10.1038/s41392-025-02269-w. URL: https://doi.org/10.1038/s41392-025-02269-w (cit. on p. 13).

[55] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. «Deep learning». In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539 (cit. on p. 13).

[56] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. «Gradient-based learning applied to document recognition». In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791 (cit. on p. 13).

[57] Y. Bengio, P. Simard, and P. Frasconi. «Learning long-term dependencies with gradient descent is difficult». In: *IEEE Transactions on Neural Networks* 5.2 (Mar. 1994), pp. 157–166. ISSN: 1941-0093. DOI: 10.1109/72.279181 (cit. on p. 13).

[58] Sepp Hochreiter and Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 13).

[59] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. *Large Language Models: A Survey*. 2025. arXiv: 2402.06196 [cs.CL]. URL: https://arxiv.org/abs/2402.06196 (cit. on p. 14).

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762 (cit. on p. 14).

[61] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. *Hyena Hierarchy: Towards Larger Convolutional Language Models*. 2023. arXiv: 2302.10866 [cs.LG]. URL: https://arxiv.org/abs/2302.10866 (cit. on pp. 15, 18).

[62] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. 2022. arXiv: 2205.14135 [cs.LG]. URL: https://arxiv.org/abs/2205.14135 (cit. on p. 15).

[63] James W. Cooley and John W. Tukey. «An algorithm for the machine calculation of complex Fourier series». In: *Mathematics of Computation* 19 (1965), pp. 297–301. URL: https://api.semanticscholar.org/CorpusID:121744946 (cit. on p. 15).

[64] Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. *Hungry Hungry Hippos: Towards Language Modeling with State Space Models*. 2023. arXiv: 2212.14052 [cs.LG]. URL: https://arxiv.org/abs/2212.14052 (cit. on pp. 16, 17).

[65] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. «DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome». In: *Bioinformatics* 37.15 (Feb. 2021), pp. 2112–2120. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab083. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/57195892/btab083.pdf. URL: https://doi.org/10.1093/bioinformatics/btab083 (cit. on p. 16).

[66] Eric Nguyen et al. *HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution*. 2023. arXiv: 2306.15794 [cs.LG]. URL: https://arxiv.org/abs/2306.15794 (cit. on p. 17).

[67] Albi Celaj et al. «An RNA foundation model enables discovery of disease mechanisms and candidate therapeutics». In: *bioRxiv* (2023). DOI: 10.1101/2023.09.20.558508. eprint: https://www.biorxiv.org/content/early/2023/09/26/2023.09.20.558508.full.pdf. URL: https://www.biorxiv.org/content/early/2023/09/26/2023.09.20.558508 (cit. on p. 17).

[68] Žiga Avsec et al. «Effective gene expression prediction from sequence by integrating long-range interactions». In: *bioRxiv* (2021). DOI: 10.1101/2021.04.07.438649. eprint: https://www.biorxiv.org/content/early/2021/04/08/2021.04.07.438649.full.pdf. URL: https://www.biorxiv.org/content/early/2021/04/08/2021.04.07.438649 (cit. on p. 17).

[69] Ken Chen, Yue Zhou, Maolin Ding, Yu Wang, Zhixiang Ren, and Yuedong Yang. «Self-supervised learning on millions of primary RNA sequences from 72 vertebrates improves sequence-based RNA splicing prediction». In: *Briefings in Bioinformatics* 25.3 (Apr. 2024), bbae163. ISSN: 1477-4054. DOI: 10.1093/bib/bbae163. eprint: https://academic.oup.com/bib/article-pdf/25/3/bbae163/57215917/bbae163.pdf. URL: https://doi.org/10.1093/bib/bbae163 (cit. on p. 17).

[70] Nicolas Scalzitti, Arnaud Kress, Romain Orhand, Thomas Weber, Luc Moulinier, Anne Jeannin-Girardon, Pierre Collet, Olivier Poch, and Julie

Thompson. «Spliceator: multi-species splice site prediction using convolutional neural networks». In: *BMC Bioinformatics* 22 (Nov. 2021). DOI: 10.1186/s12859-021-04471-3 (cit. on p. 17).

[71] Ali Saberi, Benedict Choi, Sean Wang, Mohsen Naghipourfar, Arsham Mikaeili Namini, Vijay Ramani, Amin Emad, Hamed S. Najafabadi, and Hani Goodarzi. «A long-context RNA foundation model for predicting transcriptome architecture». In: *bioRxiv* (2024). DOI: 10.1101/2024.08.26.609813. eprint: https://www.biorxiv.org/content/early/2024/08/27/2024.08.26.609813.full.pdf. URL: https://www.biorxiv.org/content/early/2024/08/27/2024.08.26.609813 (cit. on pp. 18, 28, 30, 39, 40, 43).

[72] Garyk Brixi et al. «Genome modeling and design across all domains of life with Evo 2». In: *bioRxiv* (2025). DOI: 10.1101/2025.02.18.638918. eprint: https://www.biorxiv.org/content/early/2025/02/21/2025.02.18.638918.full.pdf. URL: https://www.biorxiv.org/content/early/2025/02/21/2025.02.18.638918 (cit. on p. 19).

[73] ArcInstitute. *OpenGenome2*. 2025. URL: https://huggingface.co/datasets/arcinstitute/opengenome2 (visited on 03/14/2025) (cit. on p. 19).

[74] Tina Lenasi, B Matija Peterlin, and Peter Dovc. «Distal regulation of alternative splicing by splicing enhancer in equine beta-casein intron 1». en. In: *RNA* 12.3 (Jan. 2006), pp. 498–507 (cit. on p. 20).

[75] Oxford Nanopore Technologies. *GridION - Oxford Nanopore Technologies*. 2025. URL: https://nanoporetech.com/products/sequence/gridion (visited on 09/28/2025) (cit. on p. 21).

[76] M E Kaighn, K S Narayan, Y Ohnuki, J F Lechner, and L W Jones. «Establishment and characterization of a human prostatic carcinoma cell line (PC-3)». en. In: *Invest Urol* 17.1 (July 1979), pp. 16–23 (cit. on p. 21).

[77] Yingdong Zhao et al. «TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository». In: *Journal of Translational Medicine* 19.1 (June 2021), p. 269. ISSN: 1479-5876. DOI: 10.1186/s12967-021-02936-w. URL: https://doi.org/10.1186/s12967-021-02936-w (cit. on p. 21).

[78] W James Kent. «BLAT—the BLAST-like alignment tool». In: *Genome research* 12.4 (2002), pp. 656–664 (cit. on p. 21).

[79] Gerardo Perez et al. «The UCSC Genome Browser database: 2025 update». en. In: *Nucleic Acids Res* 53.D1 (Jan. 2025), pp. D1243–D1249 (cit. on p. 21).

[80] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. «Basic local alignment search tool». en. In: *J Mol Biol* 215.3 (Oct. 1990), pp. 403–410 (cit. on p. 22).

[81] GENCODE. *GENCODE - Homepage*. 2018. URL: https://www.gencodegenes.org/ (visited on 09/15/2025) (cit. on p. 22).

[82] Wellcome Sanger Institute. *Manual Annotation*. 2005. URL: https://www.sanger.ac.uk/collaboration/manual-annotation/ (visited on 09/15/2025) (cit. on p. 22).

[83] European Molecular Biology Laboratory's European Bioinformatics Institute. *Ensembl genome browser 115*. 2025. URL: https://www.ensembl.org/index.html (visited on 09/15/2025) (cit. on p. 22).

[84] Cold Spring Harbor Laboratory - Siepel Lab. *PHAST*. 2019. URL: http://compgen.cshl.edu/phast/ (visited on 09/15/2025) (cit. on p. 23).

[85] Jiuyong Xie, Lili Wang, and Ren-Jang Lin. «Variations of intronic branchpoint motif: identification and functional implications in splicing and disease». In: *Communications Biology* 6.1 (Nov. 2023), p. 1142. ISSN: 2399-3642. DOI: 10.1038/s42003-023-05513-7. URL: https://doi.org/10.1038/s42003-023-05513-7 (cit. on p. 26).

[86] UCSC Genome Browser. *Genome Browser IUPAC Codes*. 2025. URL: https://genome.ucsc.edu/goldenPath/help/iupac.html (visited on 08/24/2025) (cit. on p. 26).

[87] Allison J Taggart, Chien-Ling Lin, Barsha Shrestha, Claire Heintzelman, Seongwon Kim, and William G Fairbrother. «Large-scale analysis of branchpoint usage across species and cell lines». In: *Genome research* 27.4 (2017), pp. 639–649 (cit. on p. 28).

[88] Biopython. *Pairwise sequence alignment — Biopython 1.86.dev0 documentation*. 2024. URL: https://biopython.org/docs/dev/Tutorial/chapter_pairwise.html (visited on 09/28/2025) (cit. on p. 29).

[89] S B Needleman and C D Wunsch. «A general method applicable to the search for similarities in the amino acid sequence of two proteins». en. In: *J Mol Biol* 48.3 (Mar. 1970), pp. 443–453 (cit. on p. 29).

[90] T F Smith and M S Waterman. «Identification of common molecular subsequences». en. In: *J Mol Biol* 147.1 (Mar. 1981), pp. 195–197 (cit. on p. 29).

[91] Osamu Gotoh. «An improved algorithm for matching biological sequences». In: *Journal of Molecular Biology* 162.3 (1982), pp. 705–708. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(82)90398-9. URL: https://www.sciencedirect.com/science/article/pii/0022283682903989 (cit. on p. 29).

[92] M.S Waterman, T.F Smith, and W.A Beyer. «Some biological sequence metrics». In: *Advances in Mathematics* 20.3 (1976), pp. 367–387. ISSN: 0001-8708. DOI: https://doi.org/10.1016/0001-8708(76)90202-4. URL: https://www.sciencedirect.com/science/article/pii/0001870876902024 (cit. on p. 29).

[93] Angana Chakraborty and Sanghamitra Bandyopadhyay. «FOGSAA: Fast Optimal Global Sequence Alignment Algorithm». In: *Scientific Reports* 3.1 (Apr. 2013), p. 1746. ISSN: 2045-2322. DOI: 10.1038/srep01746. URL: https://doi.org/10.1038/srep01746 (cit. on p. 29).

[94] Kishore Jaganathan et al. «Predicting Splicing from Primary Sequence with Deep Learning». In: *Cell* 176.3 (Jan. 2019), 535–548.e24. ISSN: 0092-8674. DOI: 10.1016/j.cell.2018.12.015. URL: https://doi.org/10.1016/j.cell.2018.12.015 (cit. on p. 30).

[95] Inc. Illumina. *Illumina/SpliceAI: A deep learning-based tool to identify splice variants*. 2025. URL: https://github.com/Illumina/SpliceAI (visited on 09/27/2025) (cit. on p. 30).

[96] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. «Perplexity—a measure of the difficulty of speech recognition tasks». In: *The Journal of the Acoustical Society of America* 62.S1 (Aug. 2005), S63–S63. ISSN: 0001-4966. DOI: 10.1121/1.2016299. eprint: https://pubs.aip.org/asa/jasa/article-pdf/62/S1/S63/11558910/s63_5_online.pdf. URL: https://doi.org/10.1121/1.2016299 (cit. on p. 32).

[97] Shengping Yang and Gilbert Berdine. «The receiver operating characteristic (ROC) curve». In: *The Southwest Respiratory and Critical Care Chronicles* 5 (May 2017), p. 34. DOI: 10.12746/swrccc.v5i19.391 (cit. on p. 34).

[98] MartinThoma cmglee. *Roc-draft-xkcd-style.svg*. 2018. URL: https://commons.wikimedia.org/wiki/File:Roc_curve.svg (visited on 09/14/2025) (cit. on p. 36).

[99] Şeref Kerem Çorbacıoğlu and Gökhan Aksel. «Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value». en. In: *Turk J Emerg Med* 23.4 (Oct. 2023), pp. 195–198 (cit. on p. 35).

[100] HPC@PoliTO. *HPC@POLITO | Home*. 2025. URL: https://www.hpc.polito.it/index.shtml (visited on 10/07/2025) (cit. on p. 72).