Application of HyenaDNA for RNA Splicing Prediction

Ella Rannon, Keren Danan, Gal Bodek[1]
Tel-Aviv University
{ellarannon, kerendanan1, galbodek}@mail.tau.ac.il

Abstract

Exon prediction is a critical task in genomics, as it enables the identification of coding regions within DNA sequences. Recent advances in Deep Learning models have been utilized for various applications in biology, and notably have been used for predicting DNA splice sites, using both Convolutional Neural Networks (CNN) and transformer architectures. However, transformers pose challenges for long DNA sequences due to their quadratic memory requirements. In this study, we investigated the use of HyenaDNA, a novel genomic foundational model, for exon prediction. While we explored various input lengths and model architectures, the models' poor performance suggests that HyenaDNA may not be suitable for this task as-is and further adaptations may be needed to enhance HyenaDNA's performance for this task. This work underscores the ongoing need for advancements in DNA sequence prediction and analysis.

1 Introduction

Living organisms use their DNA sequences to encode instructions for protein synthesis. This DNA language is composed of nucleotides, that can consist one of four nitrogenous bases: adenine (A), cytosine (C), thymine (T), or guanine (G). In shorthand, we commonly use the notation 'A', 'C', 'T', and 'G' to represent a DNA sequence. These DNA sequences are then translated into proteins. This translation process involves two key steps: transcription of DNA into messenger RNA (mRNA) followed by the translation of mRNA into a protein. This entire process is known as the central dogma of molecular biology. Genes are comprised of coding sequences called exons and non-coding sequences called introns. RNA splicing (Berget et al., 1977) involves the removal of introns, from the precursor mRNA, while keeping the exons. This process results in a mature mRNA which is then translated into protein. The precision of this process is critical, as errors can lead to genetic disorders and diseases (Faustino and Cooper, 2003). Alternative splicing is a phenomenon where different combinations of exons within a gene can be selected during splicing, leading to multiple mRNA variants and diverse proteins (Fig.1) (Pan et al., 2008). This process greatly increases the complexity and diversity of the set of proteins produced in an organism and consequently affects the traits of the living organism. Therefore, the identification of splicing sites and variants within the genome holds significant importance. Recent developments in the field of Natural Language Processing (NLP) have used the application of NLP models to genomic data analysis, treating DNA sequences as if they were a natural language (Ng, 2017; Ji et al., 2021; Dalla-Torre et al., 2023; Zvyagin et al., 2022). The most popular architecture for tackling NLP challenges today is the Transformers architecture Vaswani et al., 2017. However, transformers require quadratic memory in respect to the input length and therefore are unsuitable for tasks that include long sequences. Since unspliced transcripts tend to be long, and due to the need for single-nucleotide resolution as even individual nucleotides affect transcript splicing, transformers may not be the best model for this task. Recently, the HyenaDNA model was introduced to address the challenge of long DNA sequences and the need for single nucleotide resolution (Nguyen et al., 2023). HyenaDNA, built upon the Hyena architecture (Poli et al., 2023), is offering the quality of attention mechanisms while reducing computational time complexity to a subquadratic level. In this work, we explore the utilization of the HyenaDNA model for the task of exon prediction, with a particular focus on the optimal sequence length for this task.

[1]with the help of Edo Dotan, a PhD student under the supervision of Prof. Tal Pupko and Dr. Yonatan Belinkov

[2]GitHub link

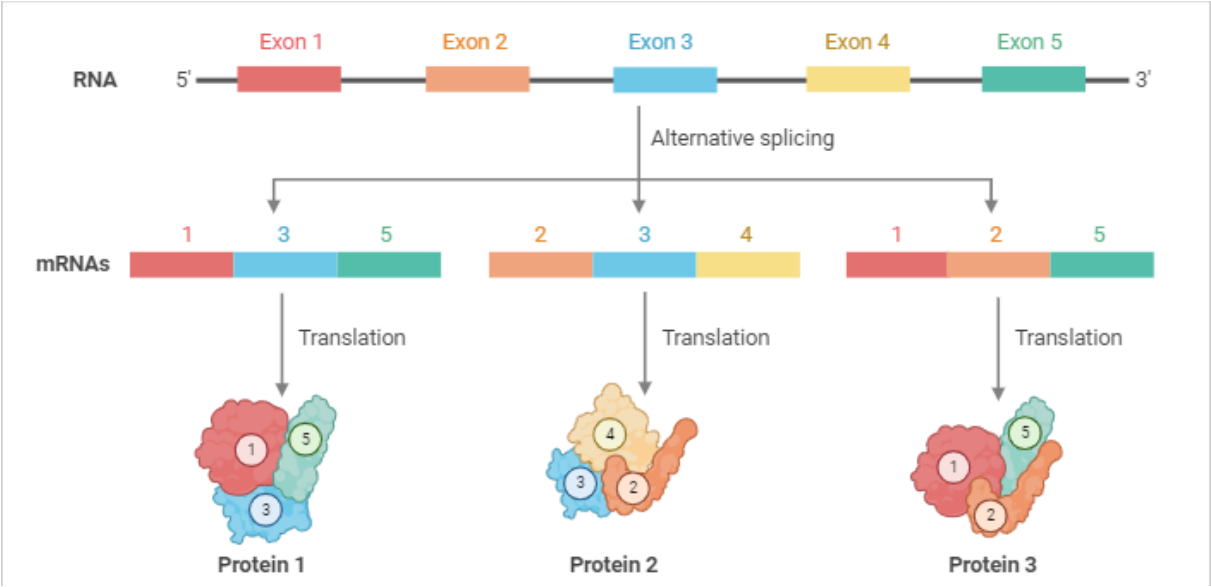

Figure 1: Alternative Splicing illustration. The RNA contains 5 exons separated by intronic regions. Then, different mRNAs can be assembled using various combinations of these exons, leading to the synthesis of distinct proteins, each with a slightly different function. Created with BioRender.com.

2 Previous Work

In recent years, various deep learning models have been employed to predict splice sites, with CNN architectures gaining popularity for this task. Notable examples are SpliceFinder (Wang et al., 2019), Spliceator (Scalzitti et al., 2021), SpliceAI (Jaganathan et al., 2019), and Splam (Chao et al., 2023). Additionally, the transformer architecture has also been explored for this task. DNABERT-SL (Leksono and Purwarianti, 2022) and the Nucleotide Transformer (Dalla-Torre et al., 2023) are two examples of works that have made use of Transformers. SpliceAI tackles splice site prediction at a per-nucleotide level using a neural network with 32 dilated convolutional layers. This model categorizes individual nucleotides into three distinct classes: donor (signifying the transition from exon to intron), acceptor (representing the transition from intron to exon), or neither. The input data for SpliceAI is derived from pre-mRNA sequences, with the nucleotide of interest positioned at the center, surrounded by a fixed number of nucleotides on both sides. Different input lengths are used during the network training; 80, 400, 2,000, and 10,000 nucleotides (nt). Remarkably, the model's performance is optimized when working with a 10,000 nt input, as it takes into account the long-range specificity conferred by the lengths of exons and introns. Similarly, SpliceFinder is trained on the human reference genome using a CNN architecture comprising two layers and a dropout layer. As seen in SpliceAI, the model receives as input sequential data with a fixed length, which is transformed into one-hot encoding, and predicts whether a given position corresponds to a splice acceptor, splice donor, or neither. During application to real genomic sequences, a sliding window is implemented in order to analyze various locations within the sequence. Similar to SpliceAI, the choice of the optimal region for training is determined by employing input sequences of different lengths ranging from 40 to 400 nt. Spliceator uses a CNN with three convolutional layers and designs different training and test datasets for multi-species splicing site prediction. Like the previous models, they train the model and evaluate its performance with different sequence lengths ranging from 20 to 600 nt. The best performance is achieved with 200 nt input sequences. Spliceator, too, employs separate models for acceptor and donor site prediction. In one of the most recent studies in this field, Splam, the researchers explore the use of notably shorter sequences compared to SpliceAI. Employing a framework based on deep convolutional neural networks, they demonstrate superior accuracy for certain explored inputs. In their model, the input consists of DNA sequences composed of 200 nt on both sides of the donor and acceptor sites, totaling 800 nt. Like the models mentioned above, the model's output provides the probability for each nucleotide, indicating whether it corresponds to a donor site, an acceptor site, or neither. In addi-

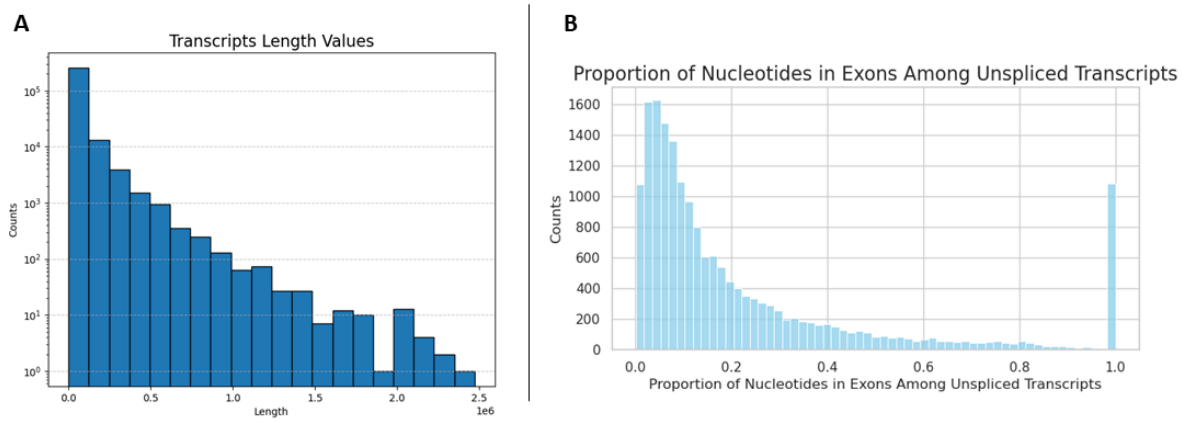

Figure 2: Analysis of the transcript sequences extracted from the Ensmbl Database. (A) Histogram describing the length of the transcripts in the data. (B) Proportion of exonic nucleotides in the transcript sequences.

tion to CNN models, the Nucleotide Transformer explores the application of transformer architectures in this context. It is benchmarked against the SpliceFinder and Spliceator models. In their study, the researchers have built four distinct foundation language models of varying sizes, and pre-trained them using three different datasets encompassing the human reference genome. For the downstream splice site prediction task, they divide the task into two parts, one following the SpliceFinder model's approach, focusing on predicting whether 400 nt sequences contain splicing sites in the center (i.e., an acceptor or donor site) or non-splicing sites within the human genome. The other part followed the Spliceator model's approach, but with the same set of sequences from Spliceator, derived from diverse organisms, where the sequences were 600 nucleotides in length. DNABERT-SL is another example of a transformer-based model for splice site prediction. It uses a pre-trained DNABERT-3 model (Ji et al., 2021), which has been trained on a large corpus of DNA sequences. The authors of the paper evaluated both fine-tuning and feature-based approaches for training the model. On validation data, the proposed model achieved high F1 scores, but its performance on test data was poor. The authors concluded that the model overfits and is therefore not yet suitable for practical use. A recent study has introduced HyenaDNA (Nguyen et al., 2023), a model based on the Hyena hierarchy (Poli et al., 2023), that was pre trained using next token predictions on human genome sequences (see Methodology 4.2). The HyenaDNA model can be then fine tuned for various downstream tasks, such as splice site prediction. The authors of the HyenaDNA paper initially addressed splice site prediction by using sequences from 100 different organisms to determine whether a sequence contains a donor or acceptor splice site.

3 Objective

In this work, we decided to focus on the HyenaDNA model. Given the somewhat abstract nature of the task defined by the HyenaDNA authors, our objective was to refine it. Rather than predicting the presence of splice sites within sequences, we aimed to predict whether the middle nucleotide belonged to an exon or an intron. This approach transformed the task into a per-nucleotide level prediction, while considering the context of surrounding nucleotides. This can allow us to employ the model for a more precise identification of exon and intron segments within new sequences. We decided to focus on experimenting with different input sizes to assess the performance of the HyenaDNA model at these varying input sizes. The selected input sizes included 400, 600, 1,000, and 10,000 nt, aligning with the lengths utilized in previous works.

4 Methodology

4.1 Data

4.1.1 Ensembl DataBase

To create our training dataset, we used the BioMart tool within the Ensembl Database, a comprehensive genomic resource (Kinsella et al., 2011). Through BioMart queries, we retrieved from the Ensembl Database data of all possible human gene transcripts, which represent different splicing options. This dataset comprises, for each transcript, two components: the sequence before splicing, containing both introns and exons, which will serve as

our model's input, and the sequence after splicing, containing only the exons, which will be used to create our model's output.

4.1.2 Preprocessing

The following approach was used for the data generation of our model input: for each transcript, we employed random sampling around each of its exons. Specifically, for a defined input length, $'L'(400, 600, 1000, 10000)$, the sampling was in the range: $(\max(0, exon\_start\_position - L)$, $\min(transcript\_length, exon\_end\_position + L))$ Within a sample of length $'L'$, the labeling of the sample corresponds to the label of the middle nucleotide, where 0 indicates this nucleotide is part of an intron and 1 indicates it is part of an exon. Furthermore, the number of sampling iterations carried out for each exon depends on the exon's length and is determined by the following formula: $transcript\_length \cdot \frac{exon\_length}{total\_exons\_length \cdot 100} \cdot$ This strategy ensured that, for longer exons, a greater amount of their contextual information was incorporated into the model inputs. After collecting all the samples, we wanted to ensure a balance between the positive (exon) and negative (intron) labels. To achieve this, we conducted undersampling of the majority class to match the size of the minority class. The data was divided into train and test sets, with chromosomes 1 and 9 assigned to the test set, and the remaining chromosomes (excluding chromosomes X, Y, and MT) assigned to the training set, as chosen in previous works. Regarding the validation set, we chose to allocate 10% of the training set, such that sequences from the same gene will not appear in both the train and validation set, to avoid leakage.

4.1.3 Data Analysis

Within the dataset we obtained from the Ensembl database, we identified a total of 70,116 genes. The average number of transcript sequences per gene is 3.9 ± 7.145. This means that, on average, each gene is associated with approximately 4 alternative splicing transcripts, which contains varying combinations of possible gene exons. The length of transcripts spans a broad range, from just a few to dozens of nucleotides, and can extend up to 2.5 million nucleotides (Fig.2A). When looking at the proportion of nucleotides within transcripts that comprise exonic regions, we can see that these exonic nucleotides typically represent a relatively small fraction of the transcript sequence, constituting less than 50% of the overall sequence (Fig.2B). There are some transcripts sequences where all of their nucleotides are a part of an exon; those are mostly short transcripts that consist entirely of one exon, thus lacking intronic sequences between exons.

4.2 Model

4.2.1 Hyena

Hyena is a large language model that offers a subquadratic time complexity and is proposed as a potential substitute for the attention mechanisms (Poli et al., 2023). Hyena is based on a class of data-controlled operators consisting of a recurrence of multiplicative gating interactions and long convolutions. Hyena can be evaluated efficiently in subquadratic time, and can learn in-context on very long sequences. Hyena was shown to match the quality of attention mechanisms while reducing computational time complexity, thereby enabling more extended contexts to be processed.

4.2.2 HyenaDNA

In this work, we utilized a recently introduced genomic foundational model named HyenaDNA (Nguyen et al., 2023) that is based on the Hyena hierarchy. HyenaDNA architecture is a simple stack of Hyena operators, pretrained on the human genome. This pretraining extended to context lengths of up to 1 million tokens, allowing for single-nucleotide resolution in the task of predicting the next nucleotide (Fig3B). The underlying concept is that the model can learn key genomic patterns during pretraining and subsequently apply this pre-trained model to address downstream biological tasks that benefit from the long-range capabilities of the HyenaDNA model. As the task of exon prediction can involve understanding the patterns within long nucleotide sequences that make up the unspliced transcripts, we believed that using the HyenaDNA model can be beneficial.

4.2.3 Fine tuning for exon prediction task

The hyenaDNA model (the medium size architecture, which was pretrained on the human reference genome at context lengths up to 160 thousand tokens) was further fine tuned for the task of exon prediction. Two distinct strategies were explored for this purpose. The first approach involved employing the classification head of hyenaDNA, while the second approach entailed the averaging of hyenaDNA's embeddings to obtain a single embedding

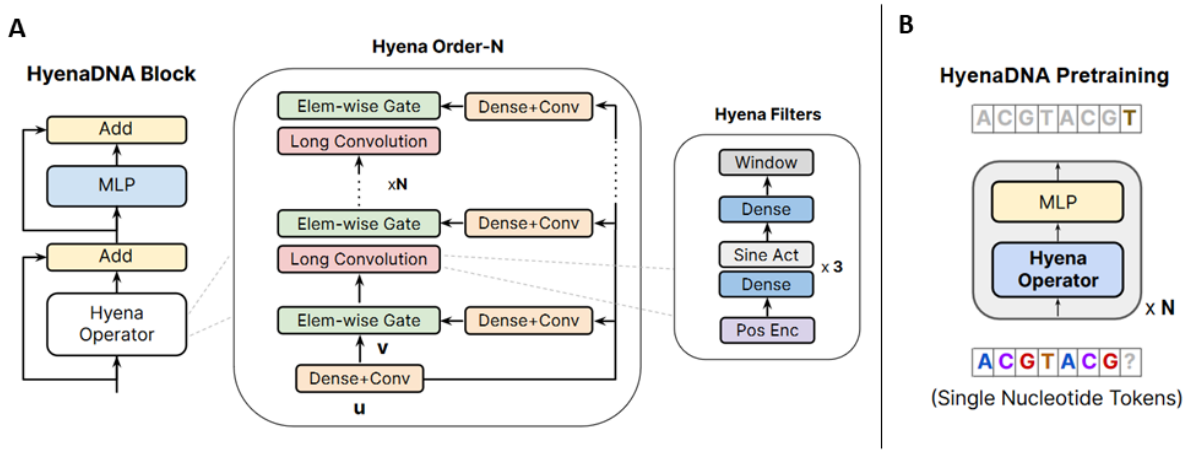

Figure 3: HyenaDNA model. (A) HyenaDNA block architecture. A Hyena operator is composed of long convolutions and elementwise gate layers. The gates are composed of projections of the input using dense layers and short convolutions. The long convolutions are parameterized implicitly via an MLP that produces the convolutional filters. (B) HyenaDNA model was pretrained using next token (nucleotide) prediction task. Source: Nguyen et al., 2023

vector for each sequence, followed by the application of dense layers to these embeddings. In both of these approaches, the training process utilized the AdamW (Loshchilov and Hutter, 2017) optimizer and employed the Cross Entropy function as the loss function. First, we compared the performance of models incorporating HyenaDNA's classification head on dataset comprising sequences of varying lengths (400, 600, 1000 and 10,000 nt). While the models trained on sequences of lengths 400 and 600 were trained using a batch size of 256, practical constraints on GPU memory necessitated smaller batch sizes for models trained on sequences of lengths 1000 and 10,000 (128 and 16, respectively). To address this, we implemented gradient accumulation (He et al., 2021), allowing model weights to be updated only after observing 256 sequences. It is worth mentioning that due to time constraints arising from the use of a small batch size for the models trained on 10,000 nt length sequences, all models in this phase were trained for a fixed duration of three epochs. Subsequently, we proceeded to evaluate the performance of models trained using the two distinct approaches on the chosen sequence length. During this phase, each model underwent training for a total of 10 epochs. In the second approach, following each layer, we employed both a non-linear activation function (specifically, Leaky ReLU (Maas et al., 2013) and dropout with a probability of 0.1. Additionally, for every layer, we decreased the dimension by a factor of two, starting with the initial embedding dimension of 256 from the HyenaDNA model and culminating with a projection layer. To assess their impact on the model's performance, several hyperparameters were examined. These hyperparameters included variations in learning rates (1e-6, 1e-5, 1e-4, 6e-4, 1e-3), different activation functions (such as ReLU Nair and Hinton, 2010, Leaky ReLU Maas et al., 2013, Sigmoid Han and Moraga, 1995, and tanh Zamanlooy and Mirhassani, 2013), and the number of dense layers (up to a maximum of 3, as depicted in (Fig.4). Regrettably, no discernible significant effects were observed as a result of these evaluations.

5 Results

5.1 Impact of Input Length on Model Performance

First, we chose the optimal sequence length for the splicing prediction task by training the hyenaDNA model on datasets of sequences of different lengths (Fig.4A), and compared the model performance on the validation set of each dataset (Table1). Since the model performance was fairly poor on all sequence lengths, we chose to focus on a single dataset and compared the performance of different models on it - the classification head of hyenaDNA, and a combination of the hyenaDNA model with a different number of dense layers (Fig.4B). The dataset we have chosen is the 600 nt, since it is the only model to predict both classes.

5.2 Impact of Model Architecture on Model Performance

We explored the influence of varying the number of layers on model performance, while maintaining a

Length	Accuracy	MCC	AUROC	AUPR	Precision	Recall	F1	Confusion matrix
400	0.5	0	0.496	0.499	0.5	1	0.66	[[0,122091], [0, 122909]]
600	0.499	-0.03	0.496	0.499	0.5	0.94	0.65	[[7026,115089], [7238, 115647]]
1000	0.5	0	0.498	0.512	1	0	0	[[119219,0], [125781, 0]]
10000	0.498	0	0.495	0.497	0.498	1	0.665	[[0, 122994], [0, 122006]]

Table 1: Performance metrics of the HyenaDNA model on a validation dataset comprised of sequences of different lengths. Confusion Matrix order: [[True Negative, False Positive], [False Negative, True Positive]]

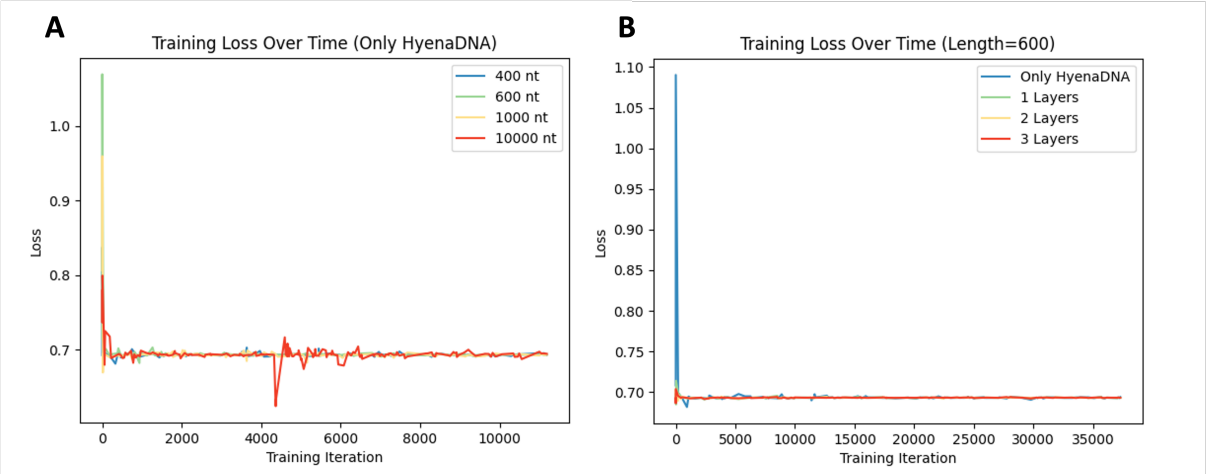

Figure 4: Training Loss Over Time. (A) Comparison of the Training Loss Over Time of Only HyenaDNA model (0 layers) for each sequence-length input; 400, 600, 1000 and 10000 nt. (B) Comparison of the Training Loss over time of each of the models for the 600 nt input length sequence.

constant input length of 600 nt. We considered four scenarios: 0 layers (utilizing only the HyenaDNA layers and classification head), 1 layer, 2 layers, and 3 layers. The outcomes are shown in (Fig.4) and Table2. Our analysis reveals that the different layers have no significant impact on performance. Nevertheless, it is worth noting that the HyenaDNA model, when devoid of additional layers, experiences more noise in the training process.

6 Discussion

The emergence of advancements in Deep Learning models in recent years have led to a wide variety of applications in biological tasks (Jumper et al., 2021; Singh et al., 2016; Li et al., 2021; Fiannaca et al., 2018). Specifically, various studies showed the effectiveness of Deep Learning models in predicting splice sites within DNA sequences (Wang et al., 2019; Scalzitti et al., 2021; Jaganathan et al., 2019; Chao et al., 2023; Leksono and Purwarianti, 2022; Dalla-Torre et al., 2023). These models primarily relied on CNN architecture, although some works also incorporated the transformer architecture, treating DNA sequences as if they were a natural language. Given the fact that DNA sequences can be very long and the need for a per-nucleotide resolution in some biological tasks, transformers might not be the most suitable architecture for this purpose. This is because transformers demand quadratic memory in relation to the input length, which can cause significant limitations. A novel genomic foundational model, HyenaDNA, has been introduced recently (Nguyen et al., 2023). It is built upon the Hyena hierarchy which has been proposed as an alternative to transformer-based models (Poli et al., 2023). It achieves attention-mechanism-quality while significantly reducing computational requirements to sub-quadratic levels in relation to input length. HyenaDNA was pre-trained on the human genome for the task of predicting the next nucleotide and can be applied to a range of downstream biological tasks. One of those tasks can be the task of exon prediction, defined in this work. The objective of our work was to investigate the effectiveness of HyenaDNA in predicting if a nucleotide is part of an intron or an exon, thereby extending the original splice site prediction task as initially defined by the authors of HyenaDNA. Their original task involved determining whether a splice site exists within a sequence, while our focus was on achieving nucleotide-level precision in exon prediction for the nucleotide of interest positioned at the center of a sequence. This involved harnessing the information from adjacent

Number of Layers	Accuracy	MCC	AUROC	AUPR	Precision	Recall	F1	Confusion matrix
Only HyenaDNA	0.5	-0.002	0.5	0.502	0.502	0.999	0.668	[[160, 121960], [180, 122700]]
1	0.5	-3.7e-5	0.499	0.501	0.502	0.415	0.454	[[71425, 50695], [71874, 51006]]
2	0.499	-0.002	0.5	0.501	0.501	0.434	0.465	[[68944, 53176], [69563, 53317]]
3	0.499	-0.003	0.498	0.501	0.499	0.259	0.341	[[90206, 31914], [91065, 31815]]

Table 2: Performance metrics of HyenaDNA model using a different number of dense layers on a test dataset (comprised of sequences of 600 nt). Confusion Matrix order: [[True Negative, False Positive], [False Negative, True Positive]]

nucleotides on both sides of the relevant nucleotide. In this specific task, we conducted experiments with varying input lengths to evaluate their impact on the model's performance. Input lengths of 400, 600, 1,000, and 10,000 nt were considered. We found that no significant impact was observed, but since the input length of 600 nt was the only one where the model managed to predict both classes, we chose to continue with this input length. Next, we investigated how different model architectures affected the results while keeping the input length fixed at 600nt. Our analysis revealed that the number of layers (0, 1, 2, and 3 layers) had no significant impact on the model's performance. However, it was evident that the model with no additional layers exhibited a noisier training process. In summary, our efforts to fine-tune HyenaDNA for the task of exon prediction on this dataset were ineffective. We can suggest some technical and biological reasons that led to the poor performance of the model. First, it is possible that the HyenaDNA, as is, lacks the ability to solve this task, and that some more adjustments need to be explored to enable the model to learn a pattern for this task. This is accentuated by the fact that all the fine-tuning tasks presented in the original HyenaDNA paper are only defined on the sequence-level, instead of nucleotide level. In addition, biological factors might have affected the model's performance. Prior research has demonstrated that the signal marking the transition from an intron to an exon, or vice versa, mostly surrounds the splice site location (Berglund et al., 1997; Mount et al., 1983; Zamore and Green, 1989). Thus, examining a wider range of the sequence exceeds the recognition capacity of the cell's splicing machinery. However, we believed that exploring longer ranges of sequences and trying to predict if a single nucleotide belongs to an intron or an exon could potentially uncover a more extensive signal than previous studies suggested. Unfortunately, our attempts to confirm this hypothesis using the HyenaDNA model did not yield conclusive results. Nevertheless, our inability to confirm our hypothesis through HyenaDNA underscores the necessity for its further exploration in future research. The work done in this study can establish a basis for future work. First, a broader range of sequence lengths can be investigated in order to determine the optimal sequence length for exon prediction and gain biological insights regarding the distance of the splicing signals. Second, for short sequences, it is worth exploring the use of a transformer architecture pre-trained on DNA sequences at the single-nucleotide level for exon prediction and comparing the performance to HyenaDNA. However, it is important to mention that a prior study, which applied a transformer pre-trained on DNA sequences at the three-nucleotide level, exhibited poor performance on a similar task (Leksono and Purwarianti 2022). In addition, it is noteworthy that in this work we averaged the embedding obtained by HyenaDNA per nucleotide to receive one embedding vector per sequence. This averaging process may have reduced the signal of the middle nucleotide. Hence, it would be valuable to explore alternatives of pooling methods or convolutions to enhance the signal captured in this context. Furthermore, it might be beneficial to explore the incorporation of residual blocks as a means to retain the global DNA signal captured by the HyenaDNA model while still adding new layers for the task of classification.

7 Limitations

There are several limitations to consider in this study. First, the use of large and lengthy data samples presented technical challenges, including memory constraints, time limitations, and resource availability and posed challenges during the training process. These constraints necessitated adjustments to the training procedure, potentially impacting the model's learning capabilities. Additionally, the random selection of data, as detailed in the methods section, might have resulted in the exclusion of nucleotides that bear specific signals marking the transition from an intron to an exon. This randomness in data selection could potentially

affect the comprehensiveness of the dataset and the model's predictive accuracy.

References

Susan M Berget, Claire Moore, and Phillip A Sharp. 1977. Spliced segments at the 5 terminus of adenovirus 2 late mrna. *Proceedings of the National Academy of Sciences*, 74(8):3171–3175.

J Andrew Berglund, Katrin Chua, Nadja Abovich, Robin Reed, and Michael Rosbash. 1997. The splicing factor bbp interacts specifically with the pre-mrna branchpoint sequence uacuaac. *Cell*, 89(5):781–787.

Kuan-Hao Chao, Alan Mao, Steven L Salzberg, and Mihaela Pertea. 2023. Splam: a deep-learning-based splice site predictor that improves spliced alignments. *bioRxiv*, pages 2023–07.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. 2023. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01.

Nuno André Faustino and Thomas A Cooper. 2003. Pre-mrna splicing and human disease. *Genes & development*, 17(4):419–437.

Antonino Fiannaca, Laura La Paglia, Massimo La Rosa, Giosue' Lo Bosco, Giovanni Renda, Riccardo Rizzo, Salvatore Gaglio, and Alfonso Urso. 2018. Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC bioinformatics*, 19:61–76.

Jun Han and Claudio Moraga. 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer.

Xiaoxin He, Fuzhao Xue, Xiaozhe Ren, and Yang You. 2021. Large-scale deep learning optimizations: A comprehensive survey. *arXiv preprint arXiv:2111.00856*.

Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F McRae, Siavash Fazel Darbandi, David Knowles, Yang I Li, Jack A Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B Schwartz, et al. 2019. Predicting splicing from primary sequence with deep learning. *Cell*, 176(3):535–548.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Rhoda J Kinsella, Andreas Kähäri, Syed Haider, Jorge Zamora, Glenn Proctor, Giulietta Spudich, Jeff Almeida-King, Daniel Staines, Paul Derwent, Arnaud Kerhornou, et al. 2011. Ensembl biomarts: a hub for data retrieval across taxonomic space. *Database*, 2011:bar030.

Muhammad Anwari Leksono and Ayu Purwarianti. 2022. Sequential labelling and dnabert for splice site prediction in homo sapiens dna. *arXiv preprint arXiv:2212.07638*.

Yu Li, Zeling Xu, Wenkai Han, Huiluo Cao, Ramzan Umarov, Aixin Yan, Ming Fan, Huan Chen, Carlos M Duarte, Lihua Li, et al. 2021. Hmd-arg: hierarchical multi-task deep learning for annotating antibiotic resistance genes. *Microbiome*, 9:1–12.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA.

Stephen M Mount, Ingvar Pettersson, Monique Hinterberger, Aavo Karmas, and Joan A Steitz. 1983. The u1 small nuclear rna-protein complex selectively binds a 5 splice site in vitro. *Cell*, 33(2):509–518.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Patrick Ng. 2017. dna2vec: Consistent vector representations of variable-length k-mers. *arXiv preprint arXiv:1701.06279*.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, et al. 2023. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15794*.

Qun Pan, Ofer Shai, Leo J Lee, Brendan J Frey, and Benjamin J Blencowe. 2008. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature genetics*, 40(12):1413–1415.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. *arXiv preprint arXiv:2302.10866*.

Nicolas Scalzitti, Arnaud Kress, Romain Orhand, Thomas Weber, Luc Moulinier, Anne Jeannin-Girardon, Pierre Collet, Olivier Poch, and Julie D Thompson. 2021. Spliceator: Multi-species splice site prediction using convolutional neural networks. *BMC bioinformatics*, 22(1):1–26.

Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. 2016. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ruohan Wang, Zishuai Wang, Jianping Wang, and Shuaicheng Li. 2019. Splicefinder: ab initio prediction of splice sites using convolutional neural network. *BMC bioinformatics*, 20:1–13.

Babak Zamanlooy and Mitra Mirhassani. 2013. Efficient vlsi implementation of neural networks with hyperbolic tangent activation function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(1):39–48.

Phillip D Zamore and Michael R Green. 1989. Identification, purification, and biochemical characterization of u2 small nuclear ribonucleoprotein auxiliary factor. *Proceedings of the National Academy of Sciences*, 86(23):9243–9247.

Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, et al. 2022. Genslms: Genome-scale language models reveal sars-cov-2 evolutionary dynamics. *bioRxiv*.

A Appendix

A.1 Prediction Task Definition

To improve our model's performance, we explored different approaches for handling the input data, considering that our data source differed from HyenaDNA's and previous studies. Our initial aim was to implement per-nucleotide classification for every nucleotide (token) within the sequence, in contrast to the HyenaDNA study, which predicted a per sequence class that determines if the sequence contains a split site or not. First, since HyenaDNA can handle long nucleotide sequences, we selected full sequences of up to 160,000 nucleotides from the data. We labeled each nucleotide as 1 or 0, depending on whether it is part of an exon or an intron. However, this dataset was imbalanced in two ways: most of the transcript nucleotides represented introns, and the transcript length distribution contained many outliers. Given the model's limited learning capacity with this data, we refined it by filtering the data to retain only the transcripts falling between the 10th and 90th percentiles in terms of their length. Next, we created a balanced dataset by employing this filter and randomly selecting exons and introns from each transcript, so that each transcript had a balanced number of nucleotides representing exons and introns. However, this adjustment did not improve the model metrics. Thus, we decided to divide the data into various sequence lengths, as observed in prior studies, while keeping the per-nucleotide prediction task. Finally, our final model was fine-tuned for exon prediction, evaluating multiple sequence lengths, but making predictions for the nucleotide positioned at the center of each sequence, as described in our methodology and results.