

ML4RG, SS24

Technische Universität München



# PROJECT IN MACHINE LEARNING FOR REGULATORY GENOMICS

*Predicting the exact Polyadenylation site by fine-tuning  
DNA language models*

HANAD ABDULLAHI, PETER NUTTER, MÅNS ROSENBAUM, SHIRLEY ZHANG

## Abstract

The genome encodes information for mRNA transcription, which then guides protein translation. After post-transcriptional processing, the mature mRNA still contains untranslated regions (UTRs) in its 5' and 3'-ends. These UTRs have many regulatory elements crucial for gene expression. Studying UTRs is challenging due to the high evolutionary rate of non-coding sequences, making sequence alignment difficult. This study aims to apply, fine-tune, and compare different DNA-language models to predict motifs in the 3'UTR, with emphasis on the polyadenylation site and related motifs. By implementing models like BPNet and DNABERT and employing fine-tuning techniques such as Low-Rank Adaptation (LoRA), we improved predictive performance. Our results show that the fine-tuned SpeciesLM model with LoRA achieved superior performance metrics, including lower validation and test loss, higher Pearson correlation, and better AUROC scores. MoDISco analysis further validated the model's ability to identify key polyadenylation motifs, demonstrating the potential of DNA language models in genomics research.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Aim of the Study . . . . .	2
<b>2</b>	<b>Polyadenylation</b>	<b>2</b>
2.1	The Polyadenylation Elements . . . . .	2
2.2	The Polyadenylation Process . . . . .	4
2.3	Sequence Element Variants and Alternate Poly(A) Signals . . . . .	4
<b>3</b>	<b>DNA Language Models</b>	<b>6</b>
3.1	Baseline model - BPNet . . . . .	6
3.1.1	Loss function . . . . .	6
3.2	Language model . . . . .	8
3.2.1	DNABERT . . . . .	8
3.2.2	DNA Species Language Model . . . . .	9
3.2.3	Metrics . . . . .	9
3.3	Finetuning . . . . .	10
3.3.1	Low Rank Adaptation (LoRA) . . . . .	10
3.4	Interpretation . . . . .	11
3.4.1	TF-MoDISco . . . . .	11
<b>4</b>	<b>Methods</b>	<b>11</b>
4.1	Data Preparation . . . . .	11
4.2	Interpretation . . . . .	12
4.2.1	TF-MoDISco . . . . .	12
<b>5</b>	<b>Results</b>	<b>12</b>
5.1	Models and Training . . . . .	13
5.2	Model Evaluation . . . . .	14
5.3	Test Metrics . . . . .	14
5.4	MoDISco Report . . . . .	14
<b>6</b>	<b>Discussion</b>	<b>17</b>
<b>7</b>	<b>Conclusions</b>	<b>17</b>
<b>8</b>	<b>Contribution statements</b>	<b>18</b>
<b>9</b>	<b>References</b>	<b>19</b>

# 1 Introduction

The genome carries the information from which mRNA can be transcribed. The mRNA is then used as a framework to translate proteins [1]. However, the majority of the DNA consists of non-coding sequences [1]. After transcription of the DNA, the introns, segments of non-coding sequences, are cleaved off in the processing of the pre-mRNA into fully functional mRNA<sup>5</sup>. However, the 5'- and the 3'-ends of the finished mRNA still consists of large, non-coding sequences. Despite not being translated, these untranslated regions (UTR) have shown to carry a plethora of regulatory elements that play a very important role in regulating the gene expression [2]. For this reason, the UTRs has become a hot-topic to study in order to understand its functions. Unfortunately, there are limitations in today's methods to study the UTRs. Non-coding sequences are more prone to evolve at a higher rate which makes today's approach sequence alignment challenging [1]. Even though the elements in the UTR involved in transcriptional and post-transcriptional control should be well preserved, orientation, arrangement and distance between elements becomes an issue [1]. That is why this field has implemented the use of DNA language models to tackle these obstacles without the need for sequence alignment.

To know the exact sequence of the UTR also requires one to know the exact boundaries for the 5'- and the 3' UTR. Focusing on the 3' end, The 3' UTR ends with the polyadenylation (poly(A)) site. In mature mRNA, a long chain of adenine bases is synthesized at the poly(A) site named the poly(A) tail, whose purpose is to stabilize the mRNA strand and protect it from degradation [3].

## 1.1 Aim of the Study

This study aims to determine where the 3' UTR ends by using different machine learning methods to predict poly(A) sites and polyadenylation elements. The aim is also to fine-tune the language model to increase its prediction accuracy. The last objective is to interpret the predictions made by the language model.

# 2 Polyadenylation

The poly(A) tail is a chain of adenine bases added in the processing stage of mRNA maturation [3]. Along with a triphosphate capping of the 5'-end, the 3'-end receives a poly(A) tail by poly(A) polymerase in the nucleus. This poly(A) tail binds to specific binding proteins to control the export the mRNA out of the nucleus into the cytoplasm, also protect the strand from being degraded and also promote translation [4].

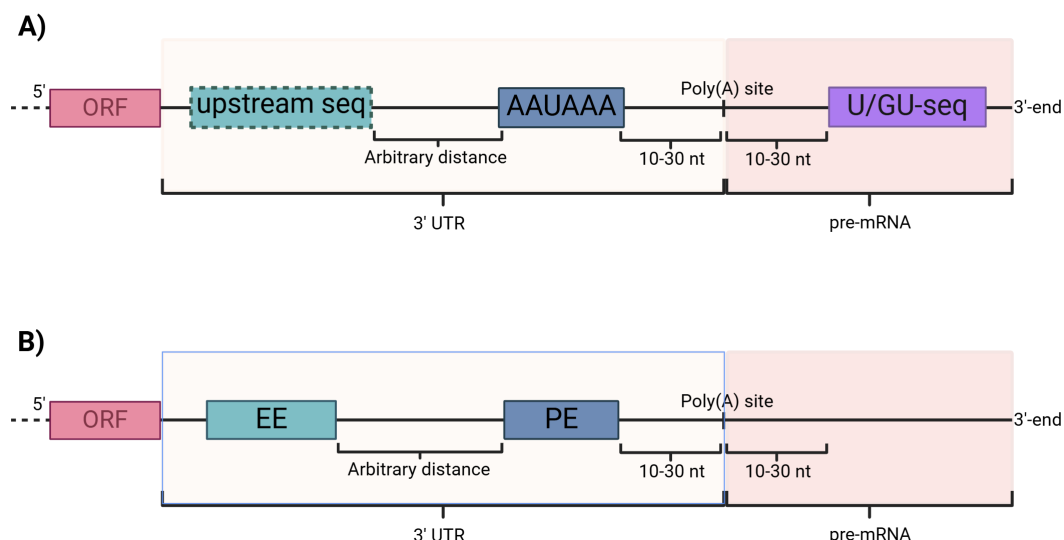
## 2.1 The Polyadenylation Elements

The requirements in forming a mature 3'-end can vary between species. The polyadenylation is controlled by the poly(A) signal which in mammalian cells is defined by three

## 2.1 The Polyadenylation Elements

---

elements in the 3' UTR [3]. These are the poly(A) site which is usually represented by the dinucleotide CA, a signaling element, usually a very conserved hexamer AAUAAA, located between 10 and 30 nucleotides upstream of the polyadenylation site, and lastly a less conserved U/GU-rich elements similarly about 10 to 30 nucleotides downstream of the poly(A) site [3, 5, 6]. The downstream elements can exist one without the other or together, with the GU-rich sequence mostly found closest to the poly(A) site [7]. Right downstream of the poly(A) site is the actual cleavage site of the pre-mRNA where the poly(A) addition will take place (Figure 1.A) [2]. The poly(A) process in yeast however, looks different. It requires three elements for functional polyadenylation, one also being the actual poly(A) site and the two other described in literature as the efficiency element and the positioning element [3, 8]. Both of these elements are located upstream of the poly(A) site, meaning polyadenylation in yeast does not depend on a downstream sequence [9]. The positioning element is 10-30 nucleotides upstream from the poly(A) sites, responsible for the actual positioning of poly(A) sites and usually consists of the same hexamer as in mammals, namely AAUAAA. While more variants exist this motif is generally very A-rich [2]. The efficiency element is located upstream of the positioning element and enhances the polyadenylation process significantly. This element is usually represented by UAUUAUA but can be more repetitive. While there can exist an enhancing sequence element in mammals too, it is not crucial and therefore not very conserved (Figure 1.B) [3]. The poly(A) site in yeast also differs from mammalian poly(A) sites, by mostly consisting of a pyrimidine (i.e. C or U) and a short stretch of A [3, 9]. In many cases, the poly(A) site is also flanked by U residues which might have a stimulating effect in cleavage-site recognition [8].



**Figure 1 – Visualization of the mRNA 3'-end in mammalian cells (A) and yeast cells (B) before polyadenylation.** **A)** For functional polyadenylation in yeast, the efficiency element is crucial. The poly(A) site consists of a C or U followed by a short stretch of A. **B)** Pre-mRNA in mammals. This process requires a U/GU-rich sequence downstream of the poly(A) site. An upstream sequence element can stimulate polyadenylation, but is not crucial. Poly(A) site usually represented by CA. ORF=open reading frame, EE=efficiency element, PE=positioning element.

## 2.2 The Polyadenylation Process

The polyadenylation process in mammalian cells is initiated by recognition of the poly(A) site in the terminal end of the 3'UTR [5]. This is fulfilled when specific cleavage factors recognise the poly(A) signal and the U/GU-rich sequence element upstream and downstream of the poly(A) site respectively. These cleavage factors, the cleavage and polyadenylation specificity factor (CPSF) and the cleavage stimulation factor (CstF) bind to the poly(A) signal (the positioning element) and the U/GU-rich element and cleaves the strand by the cleavage site. Sequentially, the poly(A) polymerase will bind to the newly cleaved 3'-end and synthesize a chain of 50 to 250 adenosine residues [5]. In yeast cells, the process is more complex but still follows the same steps: recognize the poly(A) site, cleave the pre-mRNA and synthesize the poly(A) tail [9]. This is done by several cleavage and polyadenylation factors working together [8]. Furthermore, the actual cleavage can occur after every adenosine residue that makes up the poly(A) site [8].

## 2.3 Sequence Element Variants and Alternate Poly(A) Signals

Over recent years, with the advancement in genomic and transcriptomic data analysis like deep sequencing, and the availability of improved bioinformatic tools, studies have shown

### 2.3 Sequence Element Variants and Alternate Poly(A) Signals

data that points to numerous element variants capable of promoting polyadenylation and 3'-end processing [7, 5]. What also pushed the advancements was the emergence of expressed sequence tags (ESTs), a method to sequence the 5' and 3' ends of single stranded cDNAs [5]. Numerous ESTs databases were established by several groups to investigate these variants [5]. Furthermore, the majority of eukaryotic genes have multiple polyadenylation signals, indicating the existence of alternative polyadenylation (APA) sites [7, 5]. This also explains that alternate splicing of pre-mRNA can result in APAs [7]. In one study by Guo et. al. [3], site-directed mutations was aimed to knock out elements in genes to observe potential poly(A) activity. It showed that even though important elements are knocked out, less effective elements demonstrated activity which means less optimal poly(A) sites can be used in its place. This confirms the overall dynamics of post-transcriptional regulation. All of these phenomena are crucial for cell-type specific gene expressions [6].

The hexamer AAUAAA, while being the most efficient positioning element for the poly(A) signal [5, 3], it is not the only canonical positioning element [7, 3]. Early studies pointed to the appearance of AUUAAA as an alternate positioning element with 80% of the efficiency as AAUAAA, and AGUAAA at 30% [7, 6]. In yeast cells, AAAAAA is also over-represented [3]. More variants with statistical over-representation have been observed but with lower capabilities [7].

In yeast, numerous efficiency element variants have also been observed [3]. Out of these variants, the UAUUAUA seems to have the greatest effect on 3'-end formation. However, not all genes seems to have this element variant but use other, less effective element variants like UUUUAUA and UAUGUA [3]. Furthermore, genes in yeast carrying these less effective variants have shown to together aggregate a strong signal [3].

The GU-rich downstream element in mammals can vary in few ways, from the most common GUGU to UGUG, UCUG, UGUC and more [7]. The U-rich element can vary in length but usually consist of at least three uracil residues [7]. See Table 1 and Table 2.

**Table 1 – Most common element variants in mammals.** The signaling element in mammals, corresponding to the positioning element in yeast, is mostly represented by the preserved sequence AAUAAA, since this is the most efficient element. CA marks the poly(A) site in mammals. The downstream element consist of a GU-rich element followed by a U-rich element, however, these do not need to be present together. The first row of each element represent the most common element in its category [7, 10].

Signaling element	Poly(A) site	Downstream element
AAUAAA	CA	GUGU and/or UUU...
AUUAAA		UGUG and/or UUU...
AGUAAA		CUGU and/or UUU...

**Table 2 – Most common element variants in yeast.** The efficiency element producing the strongest signal is UAUAUA. The consensus for the most common positioning element is, as in mammals, the hexamer AAUAAA. The poly(A) site is usually represented by a pyrimidine followed by a short stretch of A:s. Unlike polyadenylation in mammals, yeast does not have use of a downstream element, but requires an efficiency element upstream of the positioning element. The first row of each element represent the most common element in its category [7, 3].

Efficiency element	Positioning element	Poly(A) site
UAUAUA	AAUAAA	C(A) <sub>N</sub>
UUUAUA	AAAAAA	U(A) <sub>N</sub>
UAUGUA	UAUAAA	

## 3 DNA Language Models

### 3.1 Baseline model - BPNet

BPNet is a convolutional neural network (CNN) designed to predict transcription factor (TF) binding profiles from DNA sequences at base resolution[11]. Its applications are broad and can be extended to predict other motifs. A unique feature of BPNet is its ability to predict binding profiles based solely on the input sequence data[11]. The specific architecture of BPNet consists of several key components, as illustrated in Figure 2.

The initial convolutional layer scans the input sequence for relevant motifs. This is followed by additional convolutional layers with increasing dilation and residual connections[11]. The dilation (number of skipped positions in the convolutional filter) of these layers is doubled at each layer. Residual connections, also known as skip connections, allow the output of a layer to be added to the output of a deeper layer, enabling the output from a layer to bypass intermediate layers[11].

BPNet’s structure is similar to ResNet but features exponential dilation in the convolutional layers[11]. This approach addresses the problem of vanishing gradients, allowing BPNet to learn from high-resolution data and provide interpretable outputs.

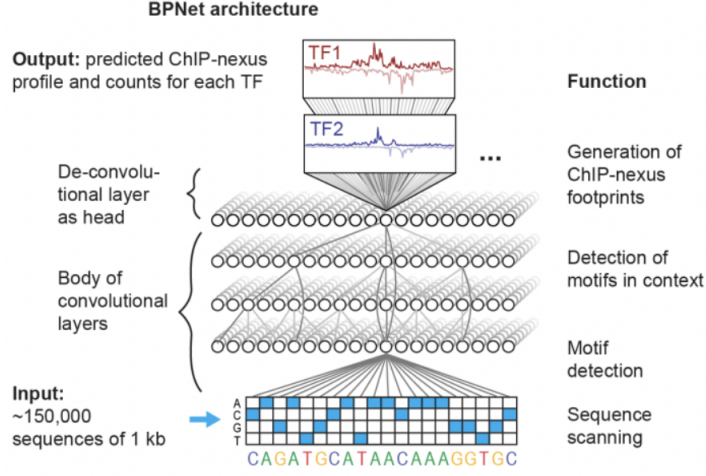
#### 3.1.1 Loss function

BPnet utilises a combination of two types of loss functions: a profile loss and a count loss. The profile loss is responsible for predicting the overall binding profile shape, while the count loss predicts the total number of reads, also known as binding intensity.

The profile loss used in BPnet is the Multinomial Negative Log-Likelihood Loss. It measures how well the predicted probability distribution of binding events matches the observed distribution across the base pairs in the sequence [11].

The count loss employed is the Mean Squared Error Loss. This loss calculates the average squared difference between the predicted and the observed total read counts, ensuring





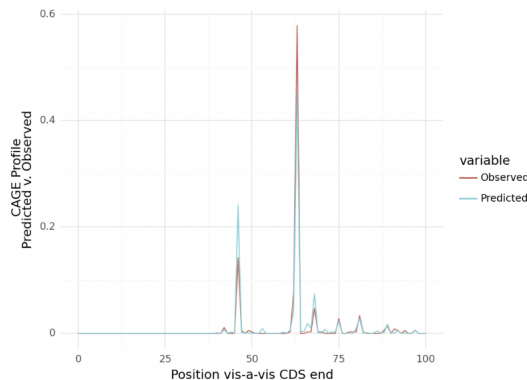
**Figure 2** – The BPNet Architecture consists of a convolutional layer that scans the input sequence for relevant motifs. The following layers are convolutional layers with increasing dilation and residual connections. The last layer is a fully connected linear layer that predicts the profile counts for each nucleotide.

that the overall magnitude of the predicted binding profile aligns with the observed binding profile [11].

Both of these loss functions are summarised in the following equation:

$$\text{Loss} = -\beta \log P_{\text{mult}}(\mathbf{k}^{\text{obs}} | \mathbf{p}^{\text{pred}}, n^{\text{obs}}) + \alpha (\log(1 + n^{\text{obs}}) - \log(1 + n^{\text{pred}}))^2$$

Here,  $\mathbf{k}^{\text{obs}}$  is the vector of length  $L$  of observed read counts for a particular strand and a particular task.  $\mathbf{p}^{\text{pred}}$  is the vector of length  $L$  of predicted probabilities along the sequence [11]. The  $n^{\text{obs}} = \sum k_i^{\text{obs}}$  is the total number of observed counts, and  $n^{\text{pred}}$  is the total number of predicted counts for the sequence. The first term represents the Multinomial Negative Log-Likelihood Loss, and the second term represents the Mean Squared Error Loss. These terms together reflect both the shape of the binding profile and the intensity [11].



**Figure 3** – The plot shows the predicted vs. observed CAGE (Cap Analysis of Gene Expression) profiles for regions 3' of annotated stop codons in *Saccharomyces cerevisiae*. The x-axis represents the position relative to the end of the Coding Sequence (CDS) while the y-axis displays the expression levels. The hyperparameter values used in the was  $\alpha = 0.2$ ,  $\beta = 0.8$ , and profile loss type was multinomial.

## 3.2 Language model

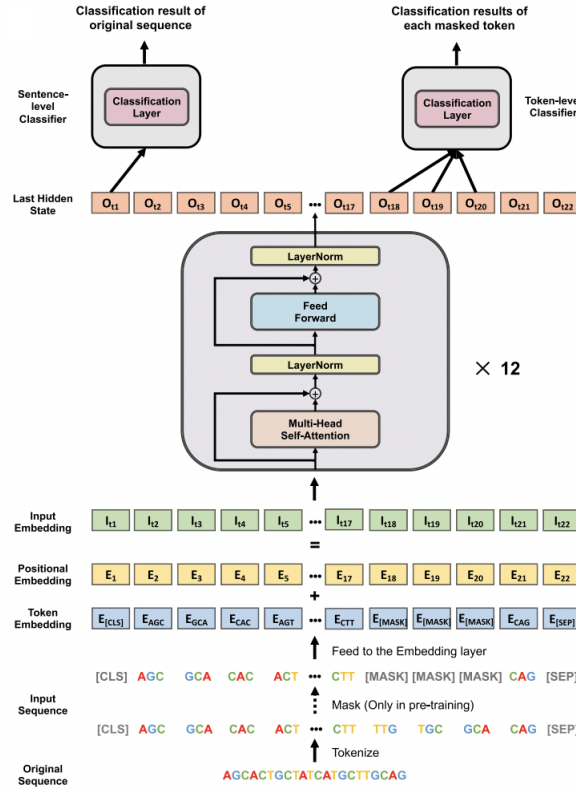
A language model is a statistical tool used in natural language processing (NLP) to predict the probability of a sequence of words. It assigns probabilities to sequences of words to predict the next word based on the preceding sequence [12]. This concept can also be applied to predicting sequences of nucleotides. Neural language models have gained popularity in recent years due to their ability to capture long-range dependencies in text and produce more accurate predictions than traditional models. Examples of neural network architectures include Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and more recently, Transformer models [12].

Recent advances in NLP due to transformers [13] have also benefited DNA sequence modeling. An example of a transformer model is BERT (Bidirectional Encoder Representations from Transformers). DNABERT is an adapted version of BERT for nucleotide sequencing, which is applied in this project.

### 3.2.1 DNABERT

DNABERT employs a transformer-based architecture tailored for genomic sequences. Unlike traditional transformer models that use an encoder-decoder structure [14], DNABERT uses only the encoder component (Figure 4). DNABERT is unique because the input DNA sequences are tokenized into overlapping k-mers, which are subsequences of length k [14]. The embedded sequences are then fed into 12 transformer layers, where the self-attention mechanism considers all input positions bidirectionally [14].

DNABERT employs a masked language modeling (MLM) approach. Randomly selected k-mers, constituting 15% of the sequence, are masked, and the model is trained to predict



**Figure 4** – DNABERT employs a transformer-based architecture. Unlike traditional transformer models that use an encoder-decoder structure, DNABERT uses only the encoder component. DNABERT is unique because the input DNA sequences are tokenized into overlapping k-mers. The embedded sequences are then fed into 12 transformer layers, where the self-attention mechanism considers all input positions bidirectionally.

these masked tokens [14]. This approach encourages the model to learn the underlying patterns and semantics of the sequences.

### 3.2.2 DNA Species Language Model

The architecture of the species-aware DNA-LM [15] used in this project is based on the DNABERT transformer model, featuring 12 encoder layers and approximately 90 million parameters. The model includes species information as an additional input token, enhancing its ability to learn and transfer regulatory features across different species.

### 3.2.3 Metrics

To evaluate the performance of our models in predicting polyadenylation (poly-A) site counts, we transform the problem into a classification task by using the softmax function. This approach allows us to leverage a range of classification metrics. The Pearson

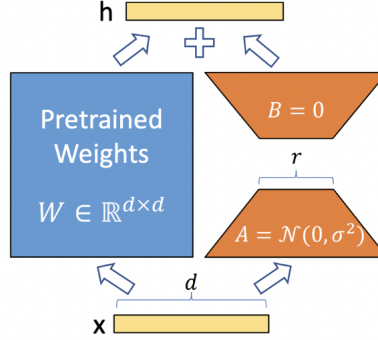
correlation coefficient measures the linear relationship between predicted and true poly-A counts. Classification metrics such as AUROC (Area Under the Receiver Operating Characteristic Curve) assess the model’s ability to correctly identify poly-A sites, focusing on precision-recall and the trade-off between true positive and false positive rates. These metrics are calculated with binning to reduce resolution and handle ambiguous positions.

### 3.3 Finetuning

A breakthrough in deep learning is the ability to inexpensively reuse pretrained foundation models and apply them to a wide range of specific tasks, a technique known as fine-tuning. Fine-tuning is employed on large language models that have been pretrained on vast amounts of data to adapt these models for specific downstream tasks. During the fine-tuning process, all parameters of interest are updated to minimize the loss on the task-specific dataset, thereby enhancing the model’s performance on that particular task. The specific Fine-tuning technique that has been applied in the project is LoRA(Low Rank Adapation).

#### 3.3.1 Low Rank Adaptation (LoRA)

LoRA is a fine-tuning technique which freezes the pre-trained model weights and introduces trainable rank decomposition matrices into each layer of the transformer architecture[16]. This because the pretrained weights have a low intrinsic rank. This means that even though the model’s pretrained weight matrices may be high-dimensional, the essential weight updates for fine-tuning lie in a lower-dimensional subspace[16]. The smaller matrices that are updated during training are derived through low rank matrix decomposition of the high-dimensional pretrained weight matrix. The benefits of using the LoRA is that it leads to fewer parameters to train and reduces computational and memory overhead. This makes also makes the adaption less resource intensive[16](Figure 5)



**Figure 5** – This Figure illustrates the Low-Rank Adaptation (LoRA) technique used to fine-tune large language models. The pretrained weights ( $W \in \mathbb{R}^{d \times d}$ ) represent the original weight matrix of the pretrained model. Two smaller matrices,  $A$  and  $B$ , are introduced to approximate the updates to  $W$ .

### 3.4 Interpretation

#### 3.4.1 TF-MoDISco

The purpose of TF-MoDISco (Transcription Factor Motif Discovery from Importance Scores) is to identify high quality non-redundant motifs from the per-base importance score of an input sequence [17]. These scores are computed using DeepLIFT, which takes the gradient of the prediction based on the input [18]. The DeepLIFT attribution scores represent the difference in the activation of the neurons between the input sequence and a reference sequence. The algorithm in TF-MoDISco involves multiple steps, which are the following: identifying important sequence segments (seqlets), clustering them into metaclusters based on their contribution scores, and merging seqlets into motifs while optimizing motif boundaries [17]. The final output of TF-MoDISco is a set of high-quality motifs that represents the recurring patterns from the input DNA sequences.

## 4 Methods

### 4.1 Data Preparation

The preprocessing of our data is essential for transforming raw genomic information into a format suitable for model training, specifically focusing on regions with poly(A) sites.

First, we load the genomic counts and TIF-seq data. The TIF-seq data is then matched with the genomic dataset to target regions around the poly-A sites. We aggregate counts of nucleotide sequences separately for glucose and galactose growth conditions, accurately aligning them to the regions of interest.

Sequences are truncated to 300 base pairs to standardize the input length for the model.

Each nucleotide is then one-hot encoded for the BPNet model or converted into 6-mers using the SpeciesLM tokenizer, with start and end padding tokens and a 'yeast' token.

The dataset is split into training, validation, and test sets based on chromosome data. Chromosomes I to XIV, and XVI are used for training, chromosome XV for validation, and chromosome VII for testing. This ensures distinct data sets for robust model evaluation.

## 4.2 Interpretation

Interpretation was done by comparing the motifs apprehended from MoDISco with literature. To further confirm credibility of the results we investigated the seqlets that MoDISco had used for creating the patterns and looked at a few of the whole sequences to find adjacent motifs in that sequence. This was also done in combination with looking at the attribution scores of these sequences.

Furthermore, all seqlets from pattern 0, 1, 2, 6, 8 and 10 was picked and searched for in their respective sequences, to detect other elements described in literature adjacent to the motifs in the seqlets.

### 4.2.1 TF-MoDISco

We have chosen DeepLIFT [18] as our method for interpretation. A technical issue arose with the way the data is tokenized when using the LM. SpeciesLM takes input IDs of the k-mers, which are then mapped using a lookup table into a 768-dimensional embedding vector. Due to the one-to-one correspondence between these entities, we take derivatives with respect to this embedding rather than the token IDs themselves. This approach allows us to obtain an attribution score for each k-mer, which we then distribute over all nucleotides corresponding to one k-mer or token. This method enables us to use a fast interpretation technique instead of relying on in silico mutagenesis. One observation is that this method leads to smoother-looking motifs in the subsequent MoDISco report. Consequently, the attribution scores and resulting motif plots must be interpreted with this method in mind, and cannot be directly compared to those produced by BPNet.

## 5 Results

To evaluate the efficacy of large language models (LLMs) for polyadenylation site prediction and LoRA for fine-tuning, we use BPNet as our baseline model. The model compared to this baseline features a mixed architecture that utilizes SpeciesLM's last hidden layer, followed by a shallow convolutional prediction network. This network is trained with low-dimensional LoRA matrices added to and trained on the LLM's encoder blocks.

To capture the effect of LoRA as a fine-tuning method through ablation, we tested multiple alternative model architectures to fairly compete with LoRA. After thorough

evaluation, we settled on a deeper convolutional network with the BPNet architecture and frozen SpeciesLM weights as an alternative to LoRA fine-tuning. This deeper network serves as a comparative model to demonstrate how allowing the signal to propagate through the entire network, instead of training on fixed embeddings, can be more effective.

We will refer to these models: model 1: BPNet , model 2: SpeciesLM + LoRA , and model 3: SpeciesLM without LoRA .

## 5.1 Models and Training

To establish fair conditions, all model hyperparameters were first optimized with a Bayesian hyperparameter search. The best hyperparameters used in the final comparison model are detailed in the following table:

Hyperparameter	BPNet	SpeciesLM + Lora	SpeciesLM
<b>Learning Rate</b>	5e-4	7e-4	5.9e-5
<b>Epochs</b>	90	65	77
<b>Params</b>	110K	92.5M	104M
<b>Trainable Params</b>	110K	2.4M	13.8M
<b>N-layers</b>	8	4	12
<b>N-filters</b>	64	[512, 256, 128]	512
<b>Batchnorm</b>	No	Yes	No
<b>Dilation</b>	Yes	No	Yes
<b>Batch size</b>	32	32	32
<b>Kernel Size [First, Middle, Last]</b>	[11, 3, 75]	[3, 3, 3]	[11, 3, 75]

**Table 3** – Model Hyperparameters

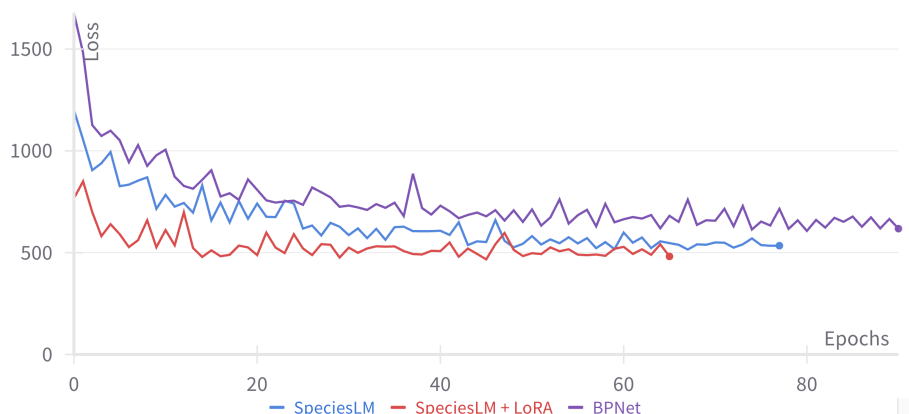
Model 1 did not benefit from increased parameter capacity; the best number of layers remained at six with a kernel size of 3. Models 2 and 3 use SpeciesLM’s last embedding layer, which is then fed into additional convolutional layers. Model 2 has 4 layers that sequentially reduce the latent space dimension from 768 to 512, 256, and 128 before the final prediction layer. This produced good results in combination with LoRA fine-tuning. Conversely, the same architecture without LoRA (with SpeciesLM weights frozen) performed below baseline, likely because the shallow trainable network couldn’t transform the embedded features into prediction counts.

To make a fair comparison to LoRA fine-tuning, we first tried unfreezing the last encoder layer of SpeciesLM to increase the network’s capacity. This increased the parameter count rapidly but produced results below baseline, likely due to the difficulty of choosing a proper learning rate for the encoder block. Our final pick for Model 3, which produced slightly better results than the baseline, was a deeper convolutional network on top of the hidden layer with the BPNet structure, using dilation in 12 layers with 256 filters. This network, although parameter-heavy, produced good results and represents an orthogonal strategy to LoRA fine-tuning.

## 5.2 Model Evaluation

The models were compared based on their validation loss. The training time for the BPNet model was significantly faster compared to Model 2 and Model 3.

We trained with a patience of 10 epochs, selecting the final models based on the best validation loss observed (6). Although the descent was not very smooth, we did not observe any double descent behavior in the larger models.



**Figure 6** – Validation loss training curves for the BPNet baseline model, the SpeciesLM model with LoRA, and the SpeciesLM model without LoRA

## 5.3 Test Metrics

All captured metrics are presented in Table 4. Model 2, which incorporates LoRA, had the lowest validation and test loss, with Model 3 trailing behind but both outperforming the baseline.

To properly assess the models, we also considered other metrics such as the mean and median Pearson correlation, providing a more intuitive measure of the models' ability to accurately capture the profile. Here again, Model 2 outperformed both compared models by a significant margin. The same trend was observed in the AUROC metric. Overall, Model 2 demonstrated clear performance gains over the baseline and Model 3 without LoRA.

## 5.4 MoDISco Report

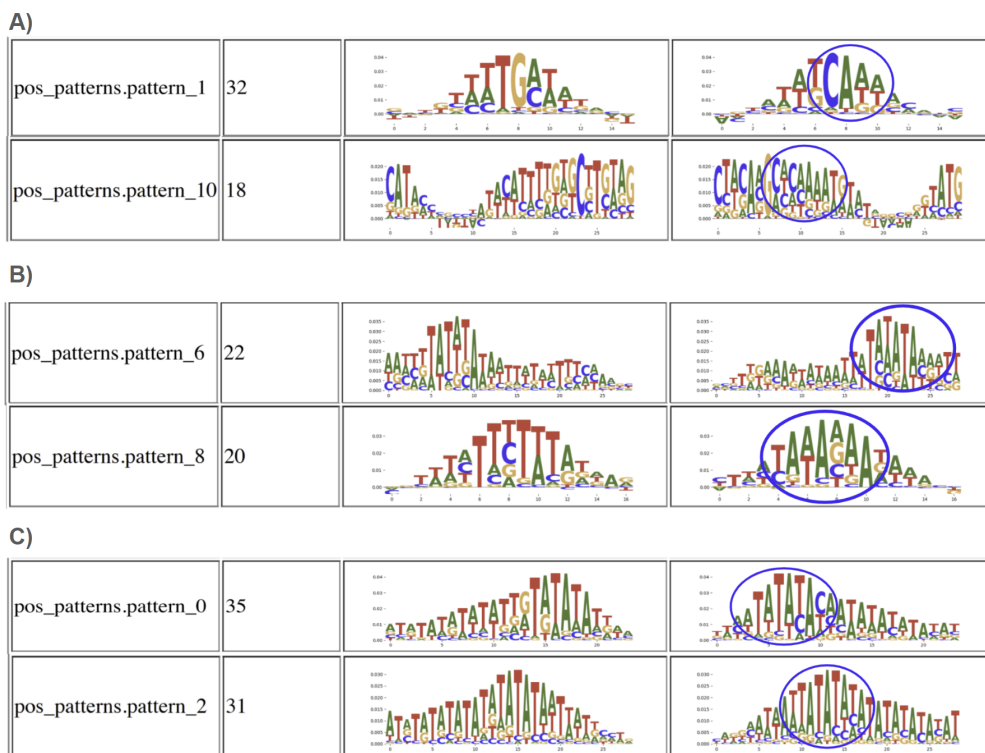
MoDISco presented 14 positive patterns for potential motifs based on the 3' species language model with LoRA finetuning. We chose six patterns in the reverse strand from the report presented below, two distinct motifs for each of the three elements found in literature (for full MoDISco report, see Appendix).



	Pearson Median	Pearson Mean	AUPRC	AUROC	Test Loss
BPNet	0.730	0.682	0.605	0.920	939.203
SpeciesLM + LoRA	0.809	0.739	0.640	0.931	711.484
SpeciesLM	0.771	0.703	0.623	0.926	844.048

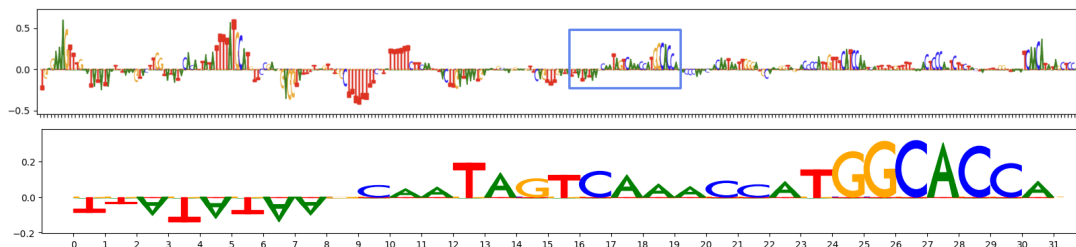
**Table 4** – Comparison of models based on test set metrics: Median and Mean Profile Pearson Correlation, Profile AUPRC, AUROC, and Loss

Looking at the results, we can see that the model with LoRA finetuning does locate motifs of interest. In figure 7, we see a clear CA dinucleotide in pattern 1 and both pattern 1 and 10 show of a short stretch of A. Additionally, both sites seems to have T:s directly upstream and downstream of the respective pattern, further indicating these could be poly(A) sites. Pattern 6 and 8 show potential positioning elements and that there might be potential variability in some bases. For example, in pattern 6 can be interpreted to include the sequence AAUAAA from positions 20 to 25, but also UAUAAA and AAAAAA (with a comparably lower probability of an A at position 23). In pattern 8, The most clear motif shows an AAAAAA positioning element from position 5 to 10, but also UAUAAA from position 4 to 9. Both patterns 0 and 2 are consistent with the efficiency element UAUAUA.



**Figure 7 – MoDISco patterns showing potential motifs for the three elements in polyadenylation in yeast.** The elements of interest are circled in blue. **A)** Two patterns of potential polyadenylation sites. Both patterns show strong indications of CA and indications of more A:s downstream. The sites in both patterns also seems to be flanked with T. **B)** Two patterns of potential positioning elements. Both patterns suggest motifs with sequences described in literature, for instance AAUAAA and UAUAAA positions 20-25 in pattern 6, and in pattern 8 UAUAAA and AAAAAA positions 4-9 and 5-10 respectively. **C)** Two patterns showing TA-rich segments which coincides with the efficiency element UAUAUA.

Since these results do not tell us about the order and spacing between motifs in the sequence, the seqlets making up the patterns (figure 7) were searched for in the sequence to look for adjacent motifs. Results show that no apparent motifs of high attribution scores can be observed (figure 8).



**Figure 8 – LoRA attribution scores for one seqlet in pattern 1.** Pattern 1 shows the motif of a poly(A) site, from which one seqlet was picked and displayed with its contribution scores showed in the lower figure. The figure above is the sequence the seqlet is derived from (blue square). The most probable poly(A) site that contributed to the motif is between position 16 and 20. Since our approach of using DeepLift allowed us to obtain attribution scores for each k-mer, which then was distributed over all nucleotides corresponding to one k-mer, this plot cannot be directly compared to the plots produced by BPNet.

## 6 Discussion

The motifs identified by MoDISco seemed to successfully find elements of interest to polyadenylation. However, these results cannot tell us about the order and spacing between motifs. In order to find any type of similarities between sequences to potentially carry a motif, MoDISco [17] searches locally for patterns of interest, hence giving a very small windows of patterns. In order to confirm that these patterns would appear in proximity to other elements, we looked at the seqlets making up each pattern 0, 1, 2, 6, 8 and 10 and searched for them in their respective sequences. After thorough investigation, we found very weak associations between the seqlets that DeepLift has provided and the sequences from the validation set of the LoRA [16] model. Furthermore, many of the seqlets found in the sequences had low and even sometimes negative contribution scores.

The reason for the disconnect between the attribution scores in the sequence on which the seqlet was derived and the final motif discovery from MoDISco could occur due to the sliding window effect and chosen sizes for the seqlets themselves.

## 7 Conclusions

This study demonstrates the effective utilization of DNA language models in predicting polyadenylation sites with single base pair accuracy. By leveraging models such as BPNet and SpeciesLM [15] and employing fine-tuning techniques like Low Rank Adaptation (LoRA) [16], we achieved significant improvements in the performance of poly(A) site predictions. Notably, the fine-tuned SpeciesLM model with LoRA outperformed baseline models across all key metrics.

MoDISco analysis validated the model’s predictive capabilities by confirming the presence of relevant motifs. The study also emphasized the critical role of efficient fine-tuning,

as the LoRA-enhanced model exhibited clear performance advantages over models using traditional fine-tuning methods. However, these performance gains over the baseline come with trade-offs, including larger model sizes, longer training times, and challenges in computing attribution scores, making them less interpretable compared to CNN kernels.

These findings highlight the potential of DNA language models in genomic research, providing a powerful tool for understanding and predicting regulatory elements in untranslated regions of mRNA. Future research could refine these models further and explore their application across various species and genomic contexts, potentially enhancing our comprehension of gene regulation and expression.

For interpretation, future research could explore alternative approaches such as *in silico* mutagenesis. It would also be beneficial to further investigate the discovery of motifs in the original sequences from which the seqlets used in MoDISco were derived.

## 8 Contribution statements

MR has done the biological research and interpretation of results. MR has written the introduction and background regarding polyadenylation.

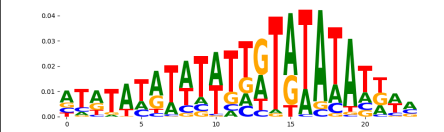
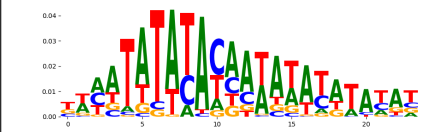
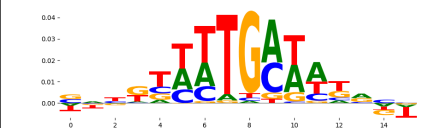
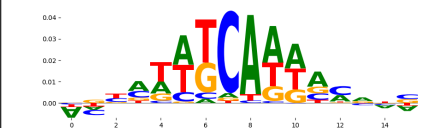
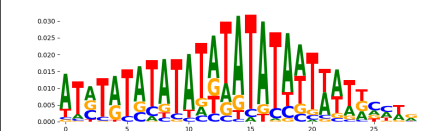
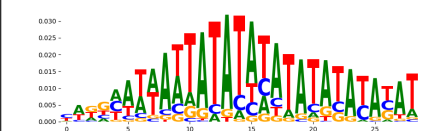
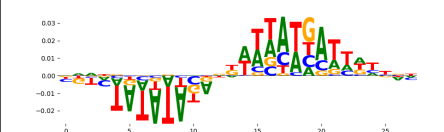
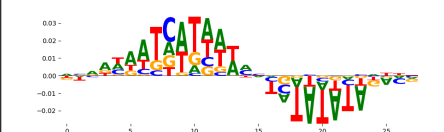
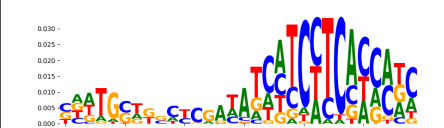



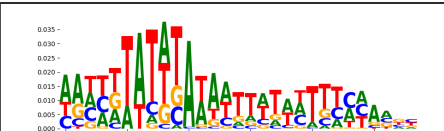
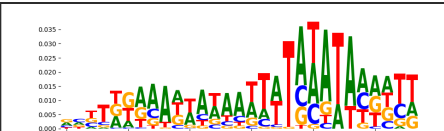






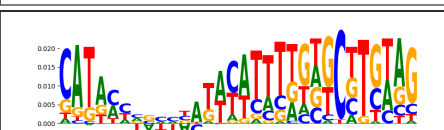
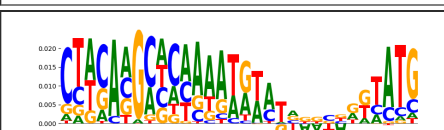
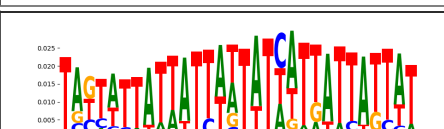
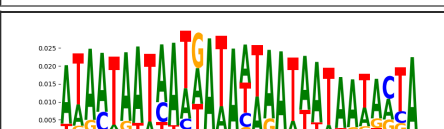
PN designed, implemented, and trained all models. Additionally, PN assisted with the attribution score and MoDIS4co generation, and authored the methods and results sections.

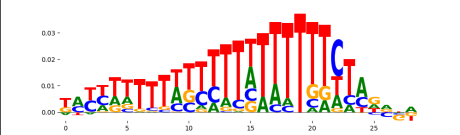
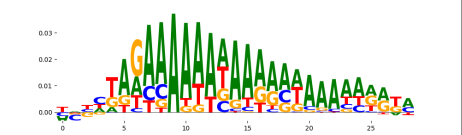
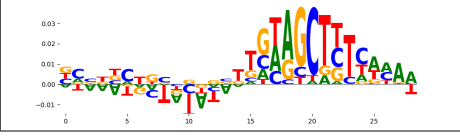
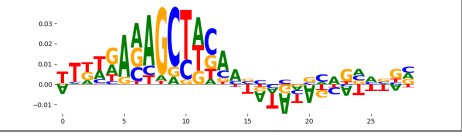
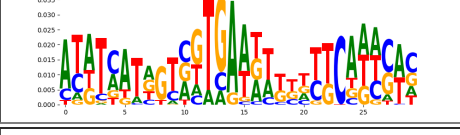
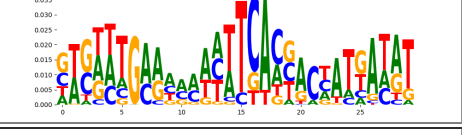
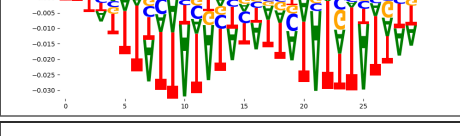
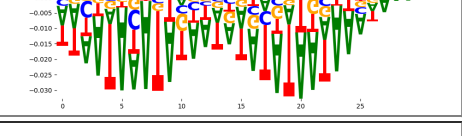
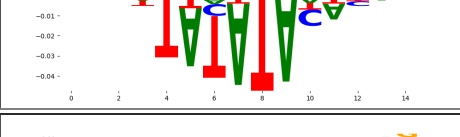
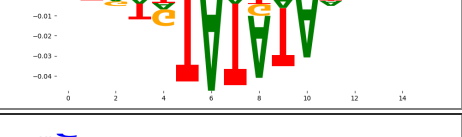
HA has interpreted the results, authored the model background, and contributed to parts of the discussion. Additionally, HA participated in providing feedback during the development of the report.

SZ has designed and implemented the interpretation methods, namely getting the attribution scores using DeepLIFT and getting the significant motifs from TF-MoDISco.

## 9 References

- [1] A. Karollus, *et al.*, *Genome Biol.* **25**, 83 (2024).
- [2] O. Shalem, *et al.*, *PLoS genetics* **11**, e1005147 (2015).
- [3] Z. Guo, F. Sherman, *Trends in biochemical sciences* **21**, 477 (1996).
- [4] L. S. Borowski, R. J. Szczesny, L. K. Brzezniak, P. P. Stepień, *Biochimica et Biophysica Acta (BBA) - Bioenergetics* **1797**, 1066 (2010). 16th European Bioenergetics Conference 2010.
- [5] C. C. MacDonald, J.-L. Redondo, *Molecular and Cellular Endocrinology* **190**, 1 (2002).
- [6] J. Neve, R. Patel, Z. Wang, A. Louey, A. M. Furger, *RNA biology* **14**, 865 (2017).
- [7] B. Tian, J. H. Graber, *Wiley interdisciplinary reviews: RNA* **3**, 385 (2012).
- [8] B. Dichtl, W. Keller, *The EMBO Journal* (2001).
- [9] E. Wahle, W. Keller, *Trends in biochemical sciences* **21**, 247 (1996).
- [10] F. Chen, C. C. MacDonald, J. Wilusf, *Nucleic acids research* **23**, 2614 (1995).
- [11] Ž. Avsec, *et al.*, *Nature genetics* **53**, 354 (2021).
- [12] W. X. Zhao, *et al.*, *arXiv preprint arXiv:2303.18223* (2023).
- [13] A. Vaswani, *et al.*, Attention is all you need (2023).
- [14] Y. Ji, Z. Zhou, H. Liu, R. V. Davuluri, *Bioinformatics* **37**, 2112 (2021).
- [15] D. Gankin, *et al.*, *bioRxiv* (2023).
- [16] E. J. Hu, *et al.*, *arXiv preprint arXiv:2106.09685* (2021).
- [17] A. Shrikumar, *et al.*, *arXiv preprint arXiv:1811.00416* (2018).
- [18] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences (2019).

pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev
pos_patterns.pattern_0	35		
pos_patterns.pattern_1	32		
pos_patterns.pattern_2	31		
pos_patterns.pattern_3	26		
pos_patterns.pattern_4	24		
pos_patterns.pattern_5	24		
pos_patterns.pattern_6	22		
pos_patterns.pattern_7	20		
pos_patterns.pattern_8	20		
pos_patterns.pattern_9	20		
pos_patterns.pattern_10	18		
pos_patterns.pattern_11	15		

pattern	num_seqlets	modisco_cwm_fwd	modisco_cwm_rev
pos_patterns.pattern_12	13		
pos_patterns.pattern_13	13		
pos_patterns.pattern_14	12		
neg_patterns.pattern_0	37		
neg_patterns.pattern_1	32		
neg_patterns.pattern_2	22	