# Joint Modelling of DNA and RNA

Final Project report - Machine Learning for Regulatory Genomics

**Date: 22.07.2025**

*Group number: 11, Group Members: Zeynep Korkmaz, Paul Kao, Valentin Ferst, Daphné Baudeu*
*Professor: Julien Gagneur, Supervisor: Tilman Hoffbauer*

## 1. Abstract

Throughout this project, we aimed to decipher the regulatory code within DNA that influences gene expression. In pursuit of this aim, we developed a CNN as well as a Transformer to jointly model DNA sequence and RNA expression. We used a masked training approach, requiring the models to infer hidden DNA nucleotides as well as masked RNA expression values from genomic context. We endeavored to identify the "grammar" learnt by our models, by identifying key DNA positions whose mutation changes the predicted RNA expression. The in silico mutagenesis analysis successfully pinpointed influential DNA regions.

## 2. Introduction

One of the main challenges in genomics has been understanding how DNA sequence drives gene expression. There is a complex network behind the regulation of RNA expression that consists of cis-regulatory elements, transcription factor binding, and chromatin accessibility. Some experimental methods give insights about these mechanisms, such as ChIP-seq, which is used to identify the specific DNA sequences that a protein binds to, or ATAC-seq, which shows which regions of DNA are open and potentially active. However, these methods are usually costly and can provide smaller-scale results. Luckily, growing computational methods and sources offer us scalable and cost-effective opportunities, where we can understand this complex network using sequence data.

It is crucial that genes are expressed at the right time, in the correct amount, and in the correct cell type. This expression is regulated through cis-regulatory elements, which are non-coding DNA sequences. These regions, such as promoters, enhancers, and silencers, serve as binding sites for transcription factors and other regulatory proteins. While promoters are found in the upstream region of the transcription site, enhancers have an impact over long distances. ~~Then~~ the DNA motifs in these elements are recognized by transcription factors, which can either activate or repress transcription. Different groups of multiple transcription factors can regulate gene expression regarding the location and number of transcripts. In conclusion, it is essential to consider the entire sequence context rather than focusing only on regions near the gene, as critical regulatory elements may be found far from the transcription start site.

In this project, we use a masked modeling framework to train two deep learning models, a CNN and a Transformer model, based on the Saccharomyces cerevisiae genome to learn dependencies between DNA sequence and RNA expression. We then evaluated our results using different evaluation methods in order to investigate how well our models capture the connection between DNA sequence and RNA expression.

## 3. Data Management and Processing

This project utilizes a multi-modal genomic dataset, structured across distinct files to support a deep learning model. Understanding how this data is organized, loaded, and split is crucial for the model's development and evaluation.

### 3.1 Data File Structure

Our primary data resides in two main files, with an optional third for context:

1. *data.npz* - This compressed NumPy archive contains the raw genomic information.
   - sequence: Stores DNA nucleotides (A, C, G, T, N) as integers (0-4), representing only the forward strand.
   - expressed_plus: Binary values (1=expressed, 0=unexpressed) for the forward DNA strand.
   - expressed_minus: Similar binary expression values, but for the backward DNA strand.
2. *regions.parquet* - This file acts as an index for data.npz. It's a table detailing genomic regions:
   - offset: The starting position of a region within data.npz.
   - contig: Identifies the chromosome (e.g., 'chrIV').
   - window_size: The length of the genomic segment.
   - strand: Indicates if the region is on the plus ('+') or minus ('-') strand, guiding data extraction and DNA sequence handling (e.g., reverse complementation for minus strand).
3. *ensembl_annotation.gff3* - An optional annotation file providing detailed biological context from Ensembl.

These files are interconnected; regions.parquet maps to specific data segments in data.npz, defining individual genomic samples.

### 3.2. Data Loading Process

The data loading process can be understood by examining the layered dataset structure we used for our Transformer Model:

1. *GenomeExpressionDataset* – It acts as the primary interface for accessing raw genomic data. It takes the "map" from regions.parquet and uses it to ~~precisely~~ locate and extract the corresponding DNA sequence and gene expression values from the large, concatenated data stored in data.npz. When a region is on the reverse strand, it adjusts the DNA sequence (reverse-complementing it) so it consistently aligns with the expression data. Essentially, it handles all the low-level data extraction and manipulation, presenting clean, ready-to-use DNA and expression pairs for the next stage of processing.
2. *MultiModalMaskingDataset* - This class processes samples from GenomeExpressionDataset. Its key functions prepare data for the model:
   - *RNA Masking* - A portion of expression labels is masked; the model predicts these hidden values.
   - *DNA Masking* - A segment of the DNA sequence is masked (e.g., replaced or randomized); the model predicts the original nucleotides. This masking process, integral to training, will be detailed in the "Methods" section.

### 3.3. Chromosome-Based Data Splitting

A crucial data preparation step is chromosome-based splitting.. This ensures no chromosome data overlaps between training, validation, and test sets, preventing data leakage and providing robust model evaluation.

A strict rule is applied: *all samples from a given chromosome reside in only one dataset.* Specifically, **Chromosomes** *IV (chrIV) and IX (chrIX) are entirely excluded from the training dataset*. This allows them to

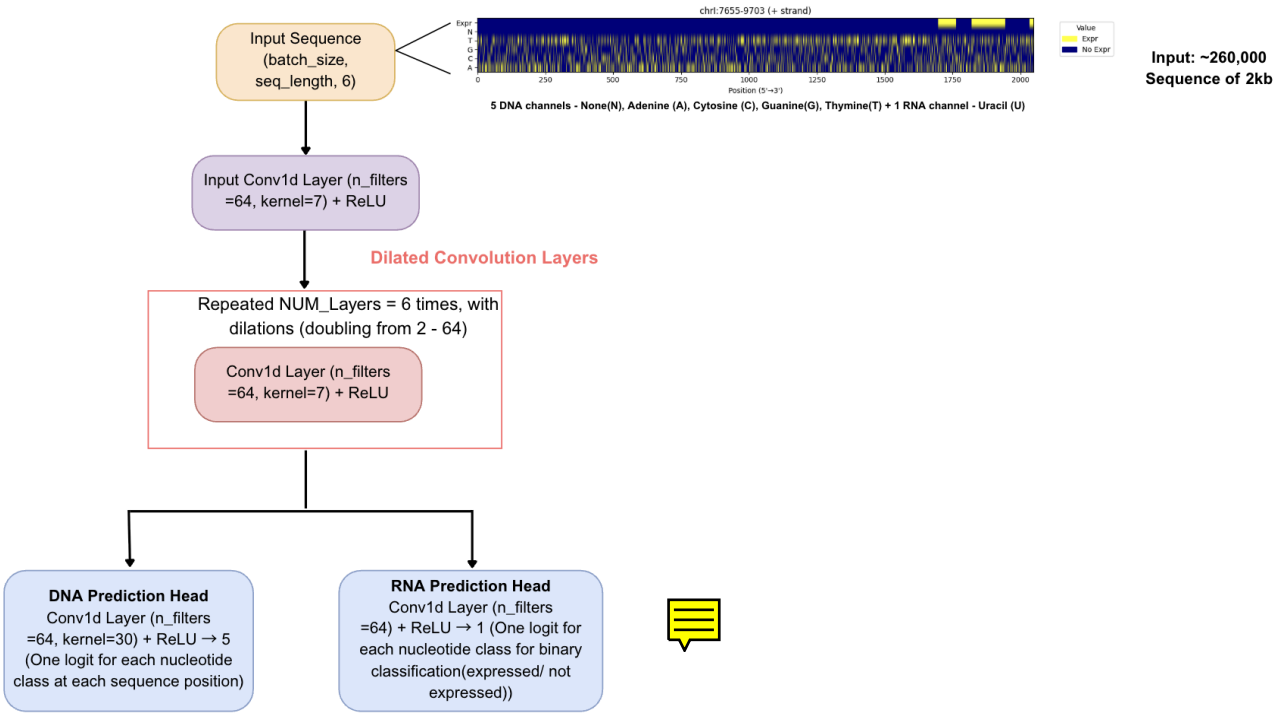
serve as dedicated test sets for later analysis, such as in-silico mutagenesis, ensuring the model's true generalization ability on entirely unseen chromosomes. Remaining chromosomes are then allocated to meet target set ratios.

## 4. Methods

### 4.1. CNN model architecture

Convolutional neural networks (CNN) have emerged as the principal tool for decoding genomic sequences by learning hierarchical, nonlinear sequence features directly from raw DNA. The convolutional filters detect short patterns in early layers and then assemble into higher-order representations when layers go deeper. However, conventional CNN architectures often produce only binary or window-averaged signals. Their reliance on pooling layers sacrifices precise motif locations and requires very deep stacks (and many parameters) to capture long-range regulatory interactions. Moreover, single-task designs focus on one output, such as transcription-factor (TF) binding, which fails to exploit shared sequence features across related genomic tasks.

BPNet sidesteps these shortcomings by removing pooling entirely and embedding an exponentially dilated residual convolutional core, which preserves base-pair resolution across kilobase contexts. Its multi-task framework simultaneously predicts per-base nucleotide identity and continuous read-count profiles, which enables shared feature representation learning that enhances both tasks and uncovers complex motif syntax.




BPNet (CNN) Model Architecture

The original BPNet model from Avsec et al., 2021 was trained to reconstruct strand-specific ChIP–nexus profiles for pluripotency factors using nine dilated residual blocks, a 25-bp first convolution and an explicit bias-control branch to factor out experimental artifacts. In contrast, our DNA–RNA BPNet adaptation begins with a Conv1D layer (6 channels, kernel size = 7), followed by ReLU, and then feeds into six residual Conv1D blocks (64 filters, kernel size = 3) with dilations doubling from 2 to 64. Finally, two parallel heads

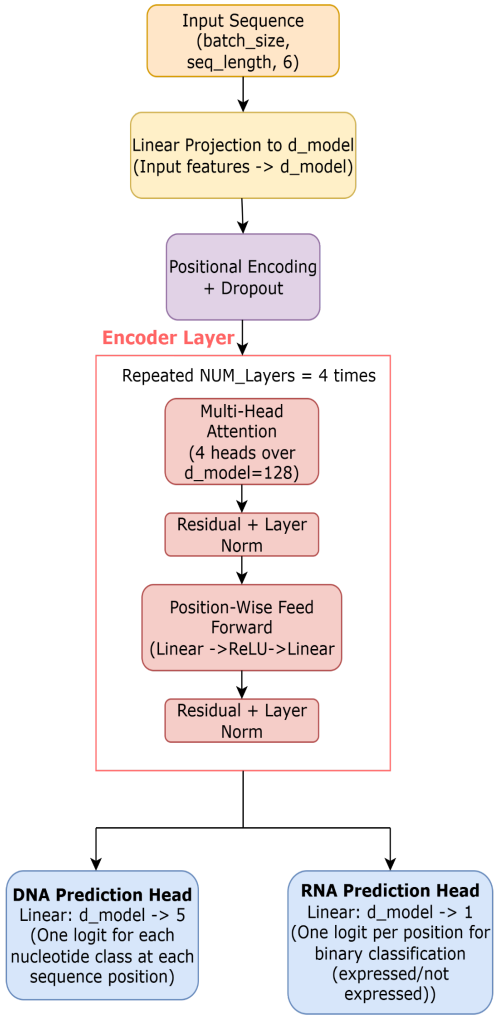
branch off: a wide-kernel (kernel size =30) Conv1D for RNA-coverage logits (expressed vs. not expressed) and a 1×1 Conv1D for five-way DNA-classification logits (A, C, G, T, N), all preserving the original sequence length. Such configuration can better balance local motif sensitivity against broad profile aggregation and remove the external bias track.

### 4.2. Transformer model architecture

In essence, the transformer takes a DNA sequence with associated RNA expression data to perform two tasks: Firstly it can predict missing DNA nucleotides that were masked from a sequence of nucleotides. Secondly, for each position in the sequence, it can predict whether the associated RNA expression is active (1) or inactive (0).

Usually, transformers consist of an Encoder layer, which uses self-attention to capture relationships between all tokens in the input sequence and generate contextual embeddings, and a Decoder layer which generates an output sequence. In our case, our transformer only consists of an Encoder layer, since a Decoder's main purpose is for creating a new sequence as an output. For our task, we are either trying to predict the type of nucleotide at masked position (not creating a full new sequence) or predicting an RNA expression which is a binary task which does not involve generating a new RNA sequence.

To summarize what the different layers of our model does, firstly we get an input sequence of the form (batch_size, seq_length, 6). The input is a batch of sequences, where each position in every sequence is described by a 6-dimensional vector representing its one-hot encoded DNA base (A, C, G, T, or N, N stands for an ambiguous nucleotide) and its associated RNA expression value.




Subsequently, a linear projection map is applied which maps input features to a vector of size d_model = 128. Then, during positional encoding we apply sinusoidal encoding to capture relative positions in the sequence.

Next comes the Encoder layer, which is repeated NUM_LAYERS= 4 times. It consists of a Multi-Head Attention with 4 attention heads, whose goal it is to simultaneously analyze relationships between different parts of the combined DNA and RNA sequence from multiple learned perspectives. Then we apply a residual connection to mitigate vanishing or exploding gradients and layer normalization to stabilize training. Subsequently we apply a position wise feed forward with a linear layer, followed by ReLU and then a linear layer again, in order to add some non-linearity and allow for more complex transformations of the contextualized embeddings produced by the attention layers. Finally, we apply a residual connection and layer normalization again.

The transformer has two output layers, each serving a distinct prediction task at each sequence position. The DNA prediction head is a linear layer which returns one out of 5 possible nucleotides (A,C,G,T,N) for each position in the input sequence. The RNA prediction head is also a linear layer that predicts a binary expression state (expressed vs. not expressed) for each nucleotide position.

### 4.3. Masking Expression and Nucleotides

To uncover intricate relationships between DNA sequence and gene expression, we used two different approaches of masking information of the genomic data.

RNA Masking: Firstly we only masked the expression labels and therefore hid a contiguous block of expression labels within each genomic window. Typically, 50% of the expression values were masked by replacing them with a specific token, which was usually set on 0.5 . This forces our model to try to predict the expression state based on the surrounding context and the corresponding DNA sequence. To enforce the model to learn a more intricate relationship between DNA sequence and RNA expression we followed a more sophisticated cross modal masking strategy. In addition to RNA Masking we now also use DNA Masking.

For DNA Masking: We employ a BERT-like strategy to hide a random subset of nucleotides. Approximately 15% of the DNA positions are selected. For these selected positions, we apply one of three possibilities: 80% of the time, the nucleotide is replaced with a generic "mask" token; 10% of the time, it is replaced with a random nucleotide; and for the remaining 10% of the time, we keep the original nucleotide.

Our models are then trained to predict the true original nucleotide for all masked positions as well as the hidden expression stats. This self-supervised task enhances our model's understanding of DNA sequence context. The MultiModalMaskingDataset of the Transformer model provides the masked input, along with the original true labels and binary masks indicating the positions that were subject to masking for both RNA and DNA, facilitating the calculation of respective loss functions during training.

## 5. Evaluation Methods

### 5.1. Sequence logos

In order to understand how certain our models are at predicting the DNA nucleotides, we plotted the sequence logos with the predicted DNA probabilities. We plotted the sequence logos for some subsegments of the two example segments of DNA from chromosome IV (positions 1,468,900 to 1,469,900) and chromosome IX (positions 47,250 to 48,250). Particularly, for the transformer, we plotted subsegments where our models were the most certain.

The sequence logos consist of a span where letters are stacked at each position. The relative sizes of the letters at each position in the sequence logo indicate the model's predicted probability for each nucleotide (A, C, G, T) at that specific position in the input DNA sequence. The height of the letters correspond to their information content. The information content at position $i$ is given by the formula:

$$IC_i = log_2(N) - H(p)_i$$

Where:

- $N$ stands for the number of possible classes a nucleotide prediction can correspond to, which is simply $N = 4$ (A,C,G,T).
- $H(p)_i$ denotes the entropy, which is a measure for quantifying the uncertainty of our predictions. It is defined by:

$$H(p)_i = -\sum_{j=1}^{N} p_{i,j} log_2(p_{i,j})$$

The term $p_{i,j}$ stands for the predicted probability that nucleotide $j$ occurs at position $i$. High entropy denotes uncertainty since it implies that the probabilities $p_{i,j}$ are evenly distributed across classes. In contrast, low entropy indicates certainty of the model, meaning that the model strongly favors one nucleotide at a specific position.

Right underneath the sequence logos, we also plotted the actual nucleotides at the corresponding positions to assess whether our models are just certain or also correct.

### 5.2. Expressed/Unexpressed Borders

The goal of the expressed–unexpressed border detector is to pinpoint, within each genomic window, the precise transitions between transcribed (expressed) and non-transcribed (unexpressed) regions. For each genomic window, we first encode the raw DNA as a one-hot tensor and use our model (BPNet / Transformer) to predict a continuous expression probability profile, $Pred(expr) \in [0,1]^L$ at every base $1, \cdots, L$. We then binarize both the true coverage and the model's prediction by thresholding at 0.5, which yields two step-functions: $\hat{y}_i, y_i \in \{0,1\}$. Contiguous runs of 1's define "expressed regions" and each transition, $1 \to 0$ or $0 \to 1$, marks a boundary. We locate these change-points by scanning for indices $i$ where $\hat{y}_i \neq y_i$, which yields the start and end coordinates of every expressed segment. Counting these segments gives both the number of expressed regions and their precise genomic boundaries.

In our combined visualization, we overlay the true and predicted step-curves with shaded bands indicating the held-out RNA-mask spans and DNA-mask spans so that one can directly see how reconstruction (masked-RNA) and full-DNA prediction interact with the boundary regions.

### 5.3. In silico mutagenesis

In-silico mutagenesis is an approach that can be used to interpret model predictions and, on a larger scale, to identify the most influential positions in a DNA sequence. In our project, we used this method to interpret the accuracy of our model predictions using our two example regions. It is important to point out that, for this evaluation, we used no masking on the DNA sequence and half masking on the RNA expression.

In this approach, for a given region of DNA sequence of length *n*, we systematically mutated every position $i \in \{1,...,n\}$ in this sequence by replacing the original nucleotide with all alternative bases $b \in \{A,C,G,T\} \setminus \{x_i\}$ one at a time. For each mutation, we recorded how the predicted RNA expression at every position *j* changed, compared to the predicted expression of the unmutated sequence. We used the absolute log odds ratio between two predictions as a comparison metric.

$$\Delta_{i,j} = \max_{b \in \{A,C,G,T\} \setminus \{x_i\}} \left| \log \left( \frac{\hat{y}_j^{(i \to b)}}{\hat{y}_j} \right) \right|$$

In our metric, $x_i$ denotes the original nucleotide at position *i*, while *b* represents a mutated nucleotide substituted at that position. $\hat{y}_j$ is the predicted RNA expression at position *j* for the original sequence, and $\hat{y}_j^{(1->b)}$ refers to the predicted expression after mutating position *i* to nucleotide *b*. The influence score, $\Delta\, i,j$, quantifies the effect of this mutation on the predicted expression.

For each position i, we took the highest value. Since the goal of this method is to reveal how strongly a nucleotide mutation influences the model's expression prediction, the maximum value at any position i represents the greatest potential impact of any mutation b. These values were then used to create a dependency matrix of size n x n. Each entry $\Delta$i,j represents the influence of mutating DNA position i on the predicted expression at RNA position j.

Using this approach on regions with known gene expression, we can evaluate whether the model is accurate in identifying the influential parts of the input region, such as the promoter site. The same approach could also be extended to a larger scale to find unknown long-range dependencies. However, this was not covered in our project due to time limitations.

### 5.4. Calibration plots

For our DNA masking strategy, it's essential to evaluate how well our model predicts different nucleotides since this helps us understand our model's strengths and weaknesses. For that, we used two different metrics. In our first metric, we calculated the predicted nucleotide distribution at the masked DNA positions and compared it to the original nucleotides. This shows us if the model is more inclined to predict specific nucleotides than others. While this is an informative method, it does not tell us how confident and well-calibrated the model is in its predictions. For example, even if the predicted distribution is close to the real distribution, we cannot analyse whether the model is overconfident in wrong predictions or underconfident in correct ones.

Therefore, in order to analyse the confidence of the model's predictions for the masked DNA bases, we used calibration plots. Here we compared the predicted probability that a masked base is a specific nucleotide with the actual fraction of times that base is correct in that probability range. For each base b∈{A, C, G, T}, predicted probabilities $\hat{p}_i^{(b)}$ were grouped into *K* equally spaced bins. For each bin *k*, we calculated:

1. The mean predicted probability in the bin:

$$\bar{p}_k^{(b)} = \frac{1}{|B_k|} \sum_{i \in B_k} \hat{p}_i^{(b)}$$

2. The true frequency for base b:

$$f_k^{(b)} = \frac{1}{|B_k|} \sum_{i \in B_k} 1_{[y_i = b]}$$

Where $B_k$ is the set of indices in bin *k*, and $1_{[y_i=b]}$ is the indicator function that equals 1 if the true base at position i is b, and 0 otherwise.

We visualized the results by plotting true frequency $f_k^{(b)}$ against $\bar{p}_k^{(b)}$ for each base. If the predicted probabilities match the true frequency, the points lie on the diagonal and we have a perfect calibration. In an overconfident model, the points lie under the diagonal line, whereas in an underconfident one, they lie over it. (3)

## 6. Results

### 6.1. CNN

#### 6.1.1. Only RNA masking: Model Accuracy & Loss

Our initial model, where we masked only half of the RNA expression window, shows the accuracy and loss trends in the *Supplement Figure 3*. In this case, we trained our model for 50 epochs on a subset of the whole dataset. As we can see in the accuracy plot, despite working on a smaller dataset, our model still managed to reach a good training accuracy of 0.74. Both training and validation accuracy increased steadily and started to plateau around epoch 40. Additionally, we observe that the validation accuracy slightly exceeded the training accuracy at certain points. Similarly, we see that both training and validation loss decreased over the 50 epochs, showing no signs of overfitting. The validation loss closely follows the training loss, indicating good generalization.

To conclude, the improved accuracy shows us that the model benefits from the increasing visibility of the DNA sequence and learns meaningful patterns without showing any under- or overfitting, as indicated by the stable gap between training and validation metrics.

### 6.1.2. RNA and DNA masking: Model Accuracy & Loss

During the 50-epoch training run, both RNA- and DNA-prediction heads improve steadily, yet with very different dynamics and ultimate performance (*Supplement Figure 4*). The overall loss curve decreases from ≈0.76 to ≈0.57, paralleled by a rise in total accuracy from ≈0.60 to ≈0.74, with training and validation tracks closely mirroring one another.

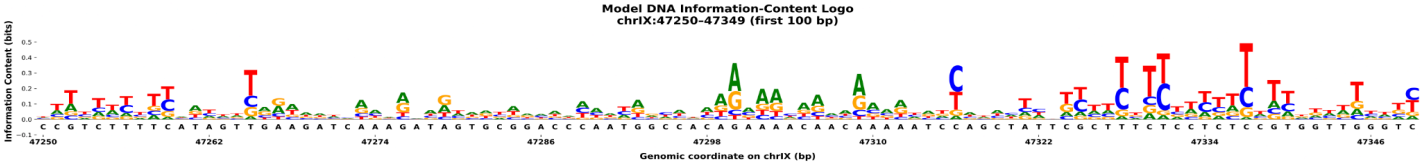
Splitting by modality, the RNA head converges quickly: its binary cross-entropy loss decreases from ≈0.58 to ≈0.35 and accuracy rises from ≈0.68 to ≈0.85, with almost overlapping train/validation lines showing little overfitting. In contrast, the DNA classifier improves more slowly: its loss decreases modestly from ≈1.34 to ≈1.30, while accuracy rises from ≈0.35 to ≈0.38. While validation performance hovers just above the training curve for DNA, suggesting stable generalization, the DNA metrics overall fall behind RNA, in line with the intrinsically harder task of nucleotide identity prediction from masked inputs.

### 6.1.3. Performance Comparison to Random Guessing

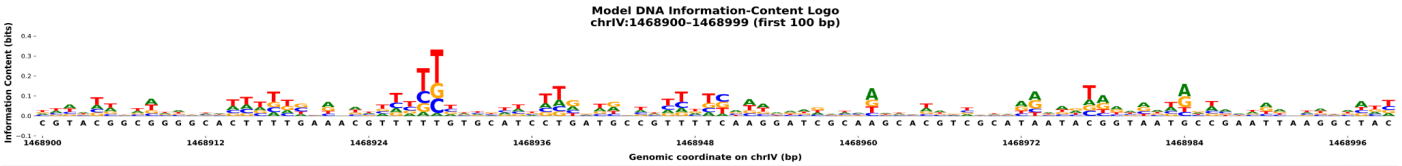
In this figure, we see a comparison between different prediction metrics of our model and the random sampling. Our BPNet model outperforms a random baseline on both DNA and RNA reconstruction tasks. For DNA, the model achieves higher accuracy (0.4233 vs. 0.2508), lower loss (1.2680 vs. 1.6460), and better F1-score and ROC AUC. The improvement is even higher for RNA, with a significantly lower loss (0.2899 vs. 1.0001), and an increase in accuracy (0.8805 vs. 0.4991), F1-score, and ROC AUC (0.9475 vs. 0.4996). (*Supplements Figure 1*)

### 6.1.4. Sequence Logos

In the first 100 bp of the chrIX window (positions 47,250–47,349), the model's information-content logo reveals several moderate-strength peaks rather than a single dominant motif. Specifically, signals of ~0.1–0.2 bits occur at ~8–12 bp and again at ~30–35 bp, with the highest peak (~0.3 bits) centered near 75 bp.



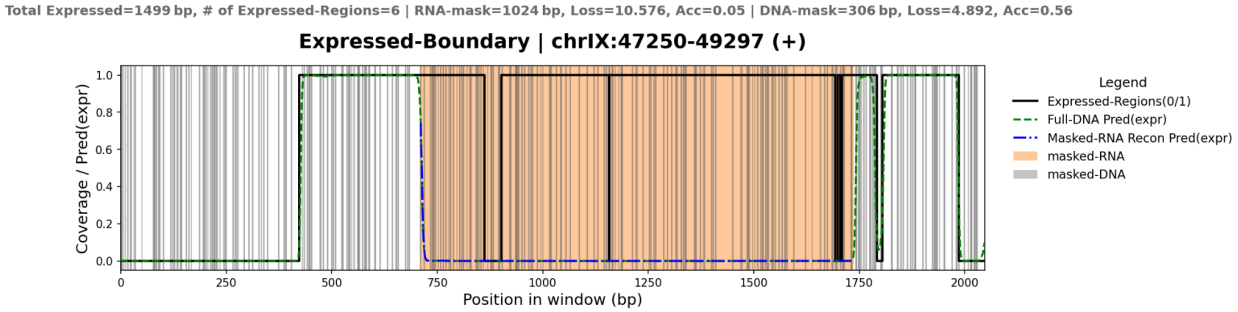

On the contrary, in the first 100 bp of the chrIV window (positions 1,468,900–1,468,999), the strongest information signal (~0.3 bits) is localized at ~25 bp, flanked by two smaller peaks (~0.1–0.2 bits) at ~15 bp and ~40 bp, after which information content gradually declines. Thus, while both regions harbor multiple informative sites, chrIX exhibits a more even dispersion of moderate-strength features, whereas chrIV is dominated by a single core motif with subsidiary flanking signals.



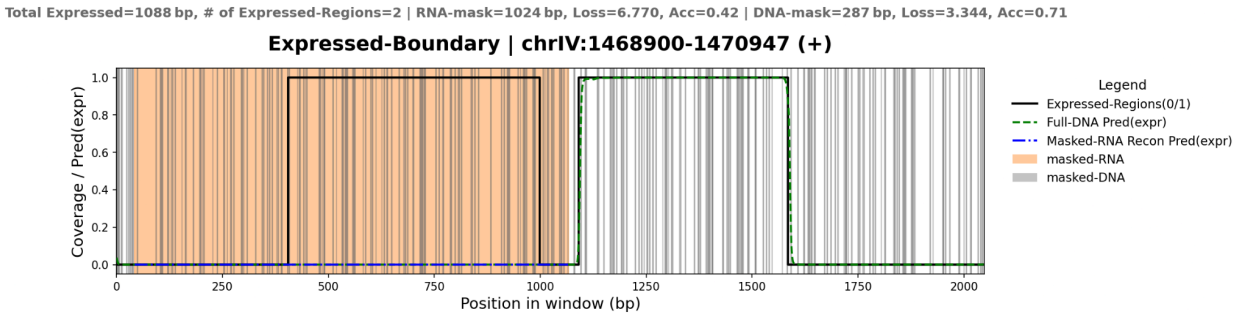

### 6.1.5. Expressed/Unexpressed Borders

Within the chrIX window (47,250–49,297 bp), the actual expression profile contains six discrete transcribed regions summing to 1,499 bp, interspersed with unexpressed gaps. When half of the RNA signal is obscured (orange shading), the model's inferred expression probability (blue dash-dot) stays close

to zero over the entire masked region (loss ≈ 9.96, accuracy ≈ 0.05), which reflects near-total failure to reconstruct the underlying step-function. Including the complete DNA sequence (green dashed) provides partial recovery (loss ≈ 4.61, accuracy ≈ 0.55), but a number of early and mid-window change-points are not consistently identified.

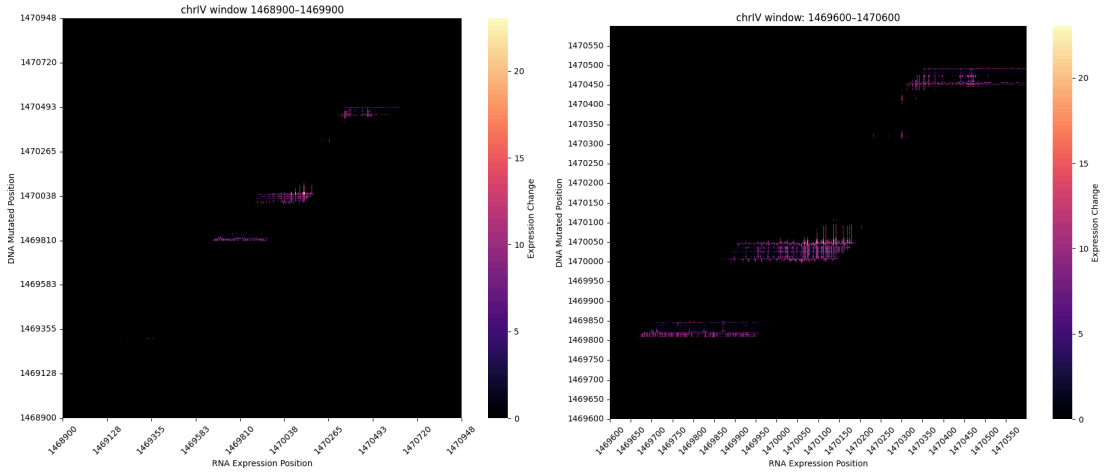



In the chrIV window (1,468,900–1,470,947 bp), a much simpler pattern of only two expressed regions (1 088 bp total) is expressed. RNA-masked reconstruction once more cannot capture the binary expression state (loss ≈ 6.41, accuracy ≈ 0.42), while full-DNA prediction again closely matches the actual step-function (loss ≈ 3.17, accuracy ≈ 0.71), which correctly delineates both the beginning and end of the single long transcribed block.

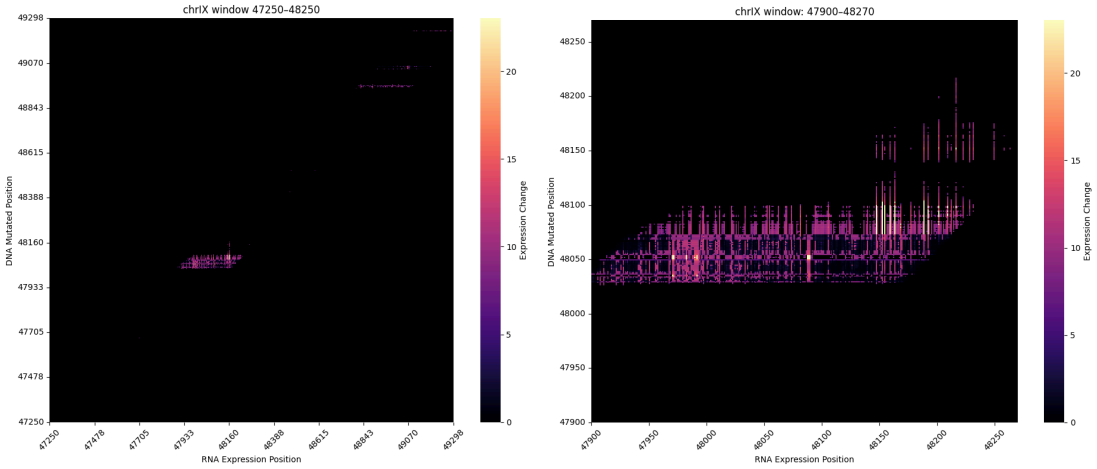



### 6.1.6. In-silico Mutagenesis

As explained in the methods section, the in-silico mutagenesis approach was applied to our two example regions to evaluate whether our model can predict the influential parts of these regions. The left side of the figure shows the calibration plot for a region that includes our first example, located on chromosome 4 between positions 1,468,900 and 1,469,900, which contains the SMT3 gene and an open reading frame. The zoomed-in plot on the right side of the figure reveals clusters of highly influential positions within this range, particularly between positions 1,460,600 and 1,470,600.



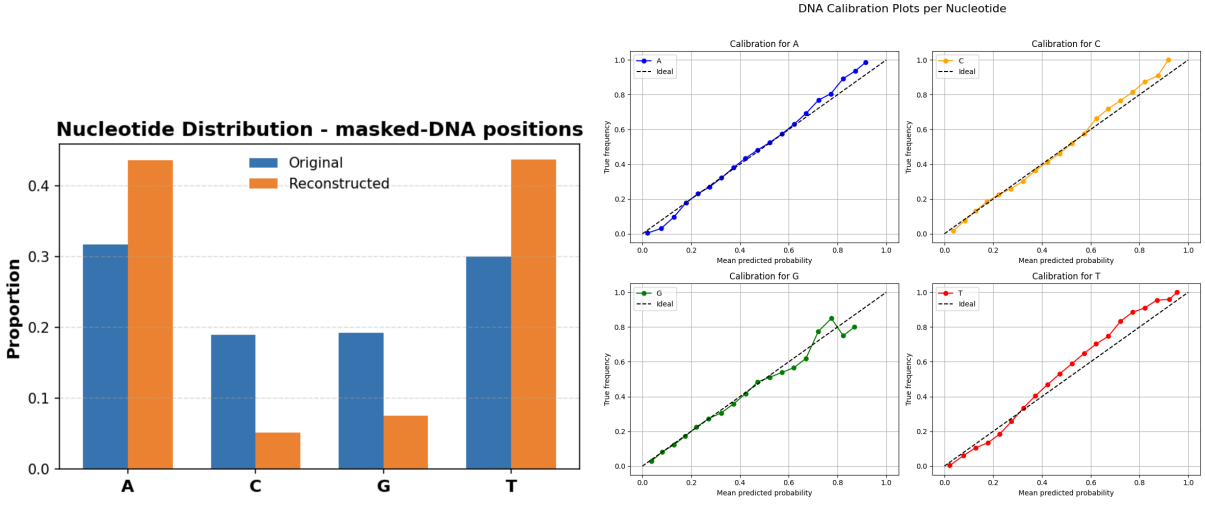
Our next example region is located on chromosome 9 between positions 47,200 and 48,250. This region contains a reverse strand gene, the *ATG4* gene with a splice site, and another dubious open reading frame. The plot on the left side of the figure shows a larger region that includes these positions. In the zoomed-in view, we observe influential regions specifically between positions 48,000 and 48,250.




### 6.1.7. Calibration Plots

As mentioned previously, we additionally analysed the nucleotide distribution at masked positions, which can be seen in figure below on the left. Here, the true nucleotide distribution is shown in blue, while the model's predicted distribution is shown in orange. We observe a strong bias toward A and T predictions, while C and G are underrepresented.

However, looking at the distribution alone is not sufficient to make inferences about the model's prediction confidence. Therefore, we created calibration plots for each nucleotide, shown in the figure on the right. In these plots, we see that A and C closely follow the ideal line, indicating good calibration. Moving on, G often shows actual precision being higher than predicted, which suggests underconfidence. T, on the other hand, appears overconfident in the high-probability bins. From a broader perspective, we can say that the calibrations deviate only slightly from the diagonal line, and the model's prediction confidence is generally acceptable, with room for improvement.



### 6.2. Transformer

#### 6.2.1. Only RNA masking: Model Accuracy & Loss

Firstly, we only masked a span of 50% of the RNA sequence to train our transformer to learn whether an RNA sequence position is expressed or not (*Supplement Figure 5)*. We only trained the model on 20.000 data points and still managed to get a test accuracy of 78.6% after 30 epochs. Similarly to the results for the CNN, we see that the model seems to neither overfit nor underfit much, as there is only a very small gap between the training and validation loss.

#### 6.2.2. RNA and DNA masking: Model Accuracy & Loss

As teased in the Methods section we now examine the training progress for our model with both RNA and DNA being masked. We therefore want to highlight two different approaches:

- **20.000 data points, 50 epochs (*Supplement Figure 6)*:** Our transformer model, trained over 50 epochs on 20,000 data points, demonstrates effective learning across both DNA and RNA modalities. The "Training and Validation Total Loss" plot shows a consistent decrease for both sets, indicating robust learning without significant overfitting. For DNA, the "Training and Validation DNA Accuracy (Masked)" plot reveals steady improvement in reconstructing masked nucleotides, with validation / accuracy closely tracking training accuracy. Similarly, the "Training and Validation RNA Accuracy" plot illustrates the model's increasing proficiency in predicting gene expression, also showing strong generalization. Finally, the test set evaluation provides quantitative confirmation of these trends, with a Test DNA Masking Accuracy of 0.3894, and Test RNA Accuracy of 0.7806.

- **100.000 data points, 9 epochs (*Supplement Figure 7)*:** Our model, trained for 9 epochs on 100,000 data points, demonstrates rapid initial learning and strong performance across both DNA and RNA masking tasks. The "Training and Validation Total Loss" plot shows a sharp decline for both sets, with the validation loss closely tracking training loss and indicating good generalization. For DNA, the "Training and Validation DNA Accuracy (Masked)" plot reveals a consistent increase in accuracy, suggesting the model quickly became proficient at reconstructing hidden DNA segments. Similarly, the "Training and Validation RNA Accuracy (Masked)" plot illustrates significant improvement in predicting gene expression, with validation accuracy aligning well with training accuracy. Finally, the test set evaluation confirms these positive trends, reporting a Test Combined Loss of 1.7325, a Test DNA Masking Accuracy of 0.3902, and a Test RNA Masking Accuracy of 0.7984.

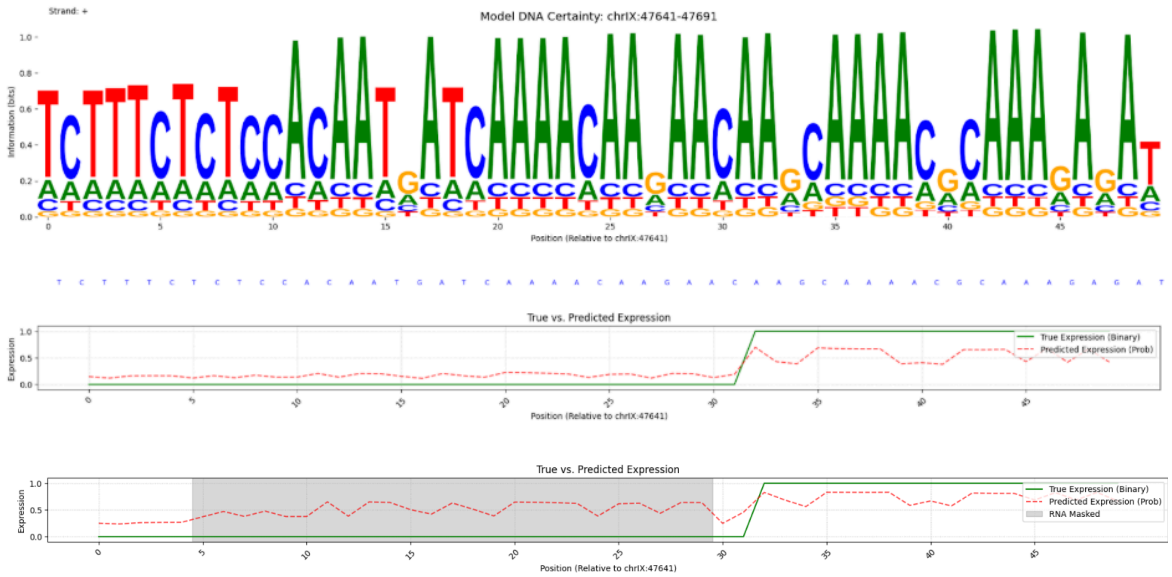
What can be deduced from these results is that more data significantly increases the learning process per epoch. After only 4 epochs of training the accuracy of the model trained with 100.000 samples can be compared with the one trained with only 20.000 datapoints.

#### 6.2.3. Performance Comparison to Random Guessing

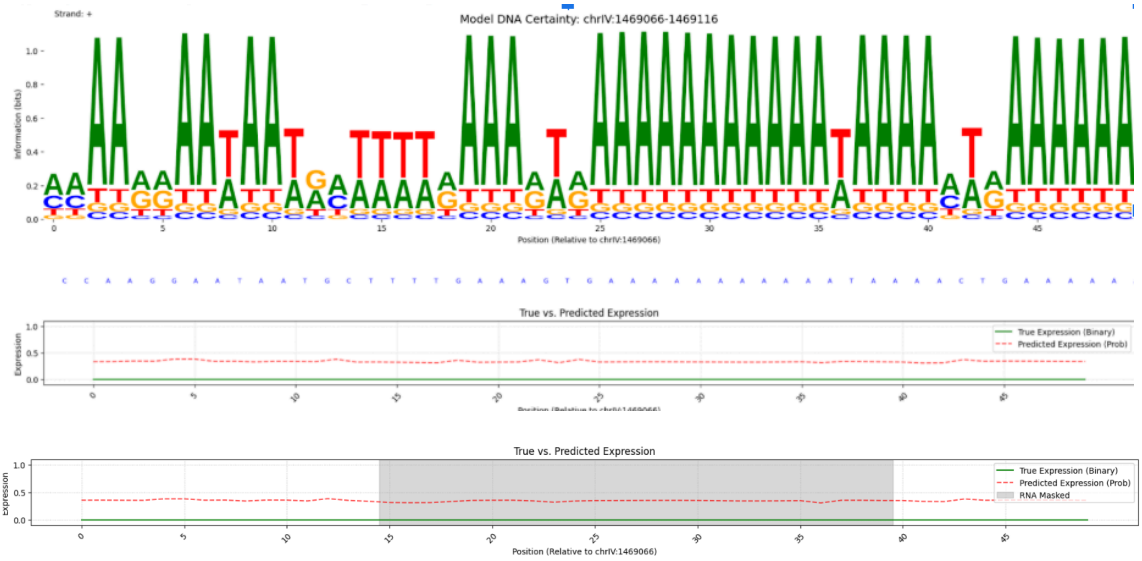
Our Transformer model consistently outperforms a stratified random baseline across all evaluated metrics on the test set, demonstrating its superior ability to learn from multi-modal genomic data. Specifically, the model achieves significantly lower losses and higher accuracy for both DNA and RNA tasks compared to random guessing, highlighted by a DNA Accuracy of 0.3910 (vs. 0.2639 for random) and an RNA Accuracy of 0.7987 (vs. 0.5087 for random). Furthermore, the substantial RNA F1-score and

impressive RNA ROC AUC of 0.8735 indicate the model's robust predictive power and generalization capabilities in gene expression prediction. (Supplement: Figure 2)

### 6.2.4. Sequence Logos and Expressed/Unexpressed Borders




*chrIX: sequence logos relative to position 47641, and expression prediction for masked and unmasked RNA sequence*




*chrIV: sequence logos relative to position 1469066, and expression prediction for masked and unmasked RNA sequence*

For both the chromosome IX (positions 47,250–48,250) and chromosome IV (positions 1,468,900–1,469,900), we visualized the sequence logos corresponding to the 50-base regions where the transformer exhibited the highest prediction certainty. Underneath that we plotted the RNA expression predictions compared to the true expressions.

On the one hand, the sequence logos show that the model displays high information content at almost every nucleotide, more so for the nucleotides A and T than the others. Indeed, as we shall see in the next section these nucleotides are overpredicted by our model. Overall, the model seems rather certain, many positions being close to 100% certainty. To verify whether our model is not only certain about its predictions but also correct, we plotted the actual ground truth sequence of nucleotides underneath the

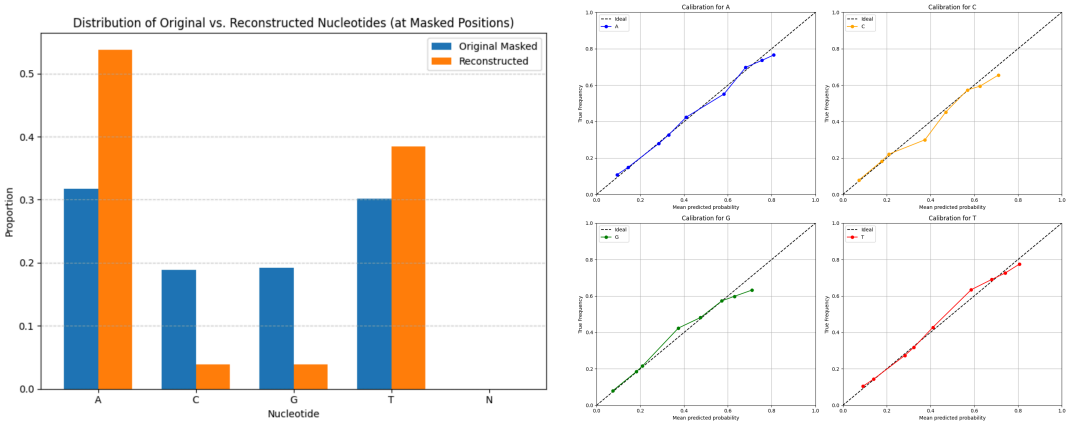
sequence logos. Comparing it to the letters the model predicts, it appears correct in most cases, especially in those where the model is most certain, indicating that the model may be well-calibrated.

On the other hand, underneath the sequence logos we plotted the model's expression predictions for the corresponding RNA regions. Firstly, we predicted without applying masking. One can observe that for the plotted region of the chromosome IX the model accurately predicts the border between expressed and unexpressed. Indeed, the predicted expression curve rises as the true expression does. The region that is plotted for chromosome IV is unexpressed, which our model also captures well. Secondly, we plotted the model's expression predictions when a span of 50% of the sequence was masked. For the region from chromosome IX we can see that the masked region's predictions are more uncertain, meaning closer to 0.5, than it previously was when unmasked. On the other hand, for the second example chromosome IV it is just as good at predicting the expression in the masked part. Overall, this suggests that while our model is learning to capture the general expression patterns and transitions, it might have weaknesses in confidently predicting unexpressed states when RNA information is missing.

To conclude, for the unmasked expression prediction task, even though the model's predicted probabilities do not consistently exhibit high confidence (i.e., values very near 0 or 1), the transformer can detect the border between an expressed and unexpressed span, and its expression predictions are mostly accurate across the two examples.

### 6.2.5. Calibration Plots

Similarly to the CNN results, we observe from the distribution of the original vs. reconstructed nucleotides that the model has a bias towards the A and the T nucleotides. This is probably simply due to the fact that they are slightly more present in the original dataset, so predicting an A or a T can easily lead to high accuracy of the model. Like for the CNN, we decided to also plot the calibration plots to see how well-calibrated our model is. Overall, the points lie close to the diagonal, especially for the nucleotides A and T. This could indicate that while the model's decision-making threshold for each class might be misaligned with the true class prevalence, the probabilities it generates are meaningful indicators of certainty.




## 7. Discussion

### 7.1. Accuracy

The accuracy metrics across our experiments provide significant insights into the capabilities of both our Transformer and CNN models under varying masking strategies. For the Transformer, training with both RNA and DNA masking consistently yielded promising accuracy trends, with both RNA and DNA validation accuracies closely tracking training accuracies, indicating strong generalization and minimal overfitting. Despite this, the Transformer's final DNA masking accuracy of approximately 0.39 on the test set, while outperforming random, suggests considerable room for improvement in reconstructing masked

DNA. This could point to the intricate nature of DNA sequence prediction or the need for more refined DNA masking techniques.

In contrast, the CNN model, particularly with RNA-only masking, demonstrated exceptional performance in predicting RNA expression, achieving a very high test accuracy of around 0.8805. Even when DNA masking was introduced, CNN's RNA accuracy remained robust, indicating its strong capacity for gene expression prediction regardless of the additional DNA task. The CNN's DNA accuracy, however, showed similar challenges to the Transformer, plateauing at around 0.4233 (from one specific comparison table, though other plots show values around 0.385).

Comparing the two architectures, both models effectively learned to predict RNA expression, with the CNN showing a slight edge in its RNA accuracy. However, accurately reconstructing masked DNA nucleotides remains a more difficult challenge for both, suggesting that this particular task is inherently more complex or requires further specialized architectural considerations. Future work could explore integrating more advanced biological priors or hybrid architectures to push the boundaries of DNA sequence reconstruction accuracy across both model types.

### 7.2. Sequence Logos

When extended to the full 2 048 bp windows, both our BPNet (CNN) and Transformer models yield information-content profiles that are dominated by A/T-rich sequence characteristics. Such pattern mirrors the abundance of polyadenylation signals (long A tracts) found in 3′-untranslated regions (UTRs) , as well as well-characterized T-rich transcription-termination motifs (e.g. "TTTATT") .

At the chrIX locus (47,500–49,547 bp, *Supplements Figure 8*), the logo presents a distributed array of moderate-strength peaks (0.1–0.3 bits) that map precisely onto the annotated 3′-UTRs of YIL157C and YIL158W, also known to harbor cleavage and polyadenylation elements responsible for appending A-rich tails to mRNAs. On the other hand, the chrIV region (1,468,900–1,470,947 bp, *Supplements Figure 8*) is largely defined by a strong A/T-rich feature (~0.5 bits) located at the 5′ end of the YDR511W open reading frame (ORF) and its adjacent 3'-untranslated region (UTR), flanked by smaller secondary peaks (~0.1–0.2 bits) that decay downstream. The corresponding pattern is representative of a unique transcription-termination/polyadenylation site, which in yeast involves the recognition of T- and A-rich sequences by the cleavage stimulation factor (CstF) and the cleavage and polyadenylation specificity factor (CPSF) complexes.

These findings suggest that our models are highly sensitive to local UTR structure and capture both general compositional biases in noncoding regions and specific polyadenylation/termination signals, instead of fitting to a solitary canonical binding motif. In future work, we plan to apply TF-MoDISco to the same hypothetical and actual contribution tracks: we anticipate its clustering of seqlets will distill the chrIV peak into a compact, high-confidence motif, while the A/T-rich clusters at chrIX will give rise to a broader ensemble of related patterns.

### 7.3. Borders Detection

In our BPNet model, RNA-masked reconstruction is generally unable to recover expression boundaries. Full-DNA prediction improves boundary detection, but its success depends strongly on the complexity of the true expression pattern; simpler two-segment loci (chrIV) yield high fidelity, whereas highly fragmented loci (chrIX) degrade performance. The transformer model seems to accurately predict expression borders, though not with very high certainty.

### 7.4. Calibration Plots

For both the CNN and the Transformer, we can remark that there seems to be a strong prediction bias towards the A and T nucleotides. This likely stems from their higher prevalence in the original dataset, leading the models to favor these nucleotides in order to attain higher overall accuracy. Despite

that, both models are well calibrated overall. Thus, whilst the models overpredict A and T due to class imbalances, their calibration plots show that when they do predict a nucleotide with high confidence they are usually correct.

### 7.5. In-silico mutagenesis

The in-silico mutagenesis results show that our CNN model can capture some biologically meaningful sequence features that influence gene expression. In the first example region on chromosome 4, although our initial focus was the SMT3 gene, the model showed strong signals around positions 1,460,600 to 1,470,600. When compared with known genomic annotations from the JBrowse Genome Browser [4], this region corresponds to the promoter of the SDH7 gene. This suggests that the model may have captured different regulatory patterns beyond the initially selected gene.

Similarly, in the second example region on chromosome 9, the model pointed to positions 48,000 to 48,250 as influential. This region aligns with the UBP7 gene according to the JBrowse Genome Browser [5], suggesting that the model managed to identify some regulatory dependencies, even though these were not directly targeted.

While these results show the model's potential in identifying both local and longer-distance sequence influences on expression, we are unsure why some of these regions were captured while others were not.

### 7.6. Conclusion

In our project, we aimed to implement joint modeling of DNA and RNA using deep learning models and have seen that it provided meaningful insights into both expression prediction and underlying sequence features. The CNN achieved high RNA prediction accuracy even with additional DNA masking, and the Transformer showed strong generalization performance. However, both of our models failed to achieve good accuracy levels for DNA reconstruction, which could prove the complexity of our goal. For future work, this suggests that the task may require more advanced modeling strategies.

One of our strongest results is the sequence logo analysis, where both of our models captured A/T-rich motifs in UTR regions that are associated with termination and polyadenylation. This means that our model doesn't only recognize standard sequence motifs but also picks up broader sequence patterns. Through the original vs. predicted nucleotide distribution plots, we have seen that the predictions are biased toward A and T nucleotides, which might be due to data imbalances. This issue could be addressed in future work. However, with our calibration plots, we have seen that the model confidence is well aligned with predictive accuracy. Finally, in-silico mutagenesis showed that the CNN can identify some regulatory regions, while failing to do so for others.

All these findings highlight the potential of deep learning models for discovering cis-regulatory relations by integrating DNA and RNA expression and point to areas for future development regarding model architecture, data bias, and interpretability.

## 8. References


1. Alharbi, W.S., Rashid, M. A review of deep learning applications in human genomics using next-generation sequencing data. *Hum Genomics* 16, 26 (2022). https://doi.org/10.1186/s40246-022-00396-x
2. Avsec, Ž., Weilert, M., Shrikumar, A. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat Genet* 53, 354–366 (2021). https://doi.org/10.1038/s41588-021-00782-6
3. Bansal, S. . *Understanding model calibration in machine learning*. *Medium*. (2024) https://medium.com/@sahilbansal480/understanding-model-calibration-in-machine-learning-6701814dbb3a
4. Taskiran, I.I., Spanier, K.I., Dickmänken, H. *et al.* Cell-type-directed design of synthetic enhancers. *Nature* 626, 212–220 (2024). https://doi.org/10.1038/s41586-023-06936-2
5. Thomas Hayes *et al.*, Simulating 500 million years of evolution with a language model. *Science* **387**,850-858(2025).DOI: 10.1126/science.ads0018
6. Karollus, A., Hingerl, J., Gankin, D. *et al.* Species-aware DNA language models capture regulatory elements and their evolution. *Genome Biol* 25, 83 (2024). https://doi.org/10.1186/s13059-024-03221-x
7. Tomaz et al. Nucleotide dependency analysis of DNA language models reveals genomic functional elements. bioRxiv 2024.07.27.605418 (2024). https://doi.org/10.1101/2024.07.27.605418
8. Shrikumar, A., Tian, K., Avsec, Ž. et al. Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5. arXiv e-prints, arXiv:1811.00416 (2018). https://doi.org/10.48550/arXiv.1811.00416
9. Cleavage and polyadenylation specificity factor: https://en.wikipedia.org/wiki/Cleavage_and_polyadenylation_specificity_factor
10. Genome Browser: https://jbrowse.yeastgenome.org/?loc=chrIV%3A1468900..1469900&tracks=DNA%2CAll%20Annotated%20Sequence%20Features%2CDoube_strand_break_hotspots%2CXrn1-sensitive_unstable%20transcripts_XUTs%2CScGlycerolMedia%2C3%27UTRs%2CPolII_occupancy_WT&highlight=
11. Genome Browser: https://jbrowse.yeastgenome.org/?loc=chrIX%3A48000..48270&tracks=DNA%2CAll%20Annotated%20Sequence%20Features%2CDoube_strand_break_hotspots%2CXrn1-sensitive_unstable%20transcripts_XUTs%2CScGlycerolMedia%2C3%27UTRs%2CPolII_occupancy_WT&highlight=

## 9. Supplement

Figure 1: CNN - Model Metrics in comparison to Random Sampling

PERFORMANCE COMPARISON

| Metric | Random | Your BPNet(CNN) Model |
|---|---|---|
| DNA Loss | 1.6460 | 1.2680 |
| DNA Accuracy | 0.2508 | 0.4233 |
| DNA F1-score (wtd) | 0.2544 | 0.3810 |
| DNA ROC AUC | 0.4999 | 0.6555 |
| RNA Loss | 1.0001 | 0.2899 |
| RNA Accuracy | 0.4991 | 0.8805 |
| RNA F1-score (wtd) | 0.5006 | 0.8806 |
| RNA ROC AUC | 0.4996 | 0.9475 |

Figure 2: Transformer - Model Metrics in comparison to Random Sampling

PERFORMANCE COMPARISON (TEST SET)

| Metric | Stratified Random | Our Transformer Model |
|---|---|---|
| DNA Loss (Cross-Entropy) | 1.3585 | 1.2863 |
| DNA Accuracy | 0.2639 | 0.3910 |
| DNA F1-score (Weighted) | 0.2641 | 0.3483 |
| RNA Loss (BCE) | 0.6838 | 0.4465 |
| RNA Accuracy | 0.5087 | 0.7987 |
| RNA F1-score (Weighted) | 0.5084 | 0.7956 |
| RNA ROC AUC | 0.5000 | 0.8735 |

Figure 3: CNN - Model Accuracy & Loss (masking RNA Only)

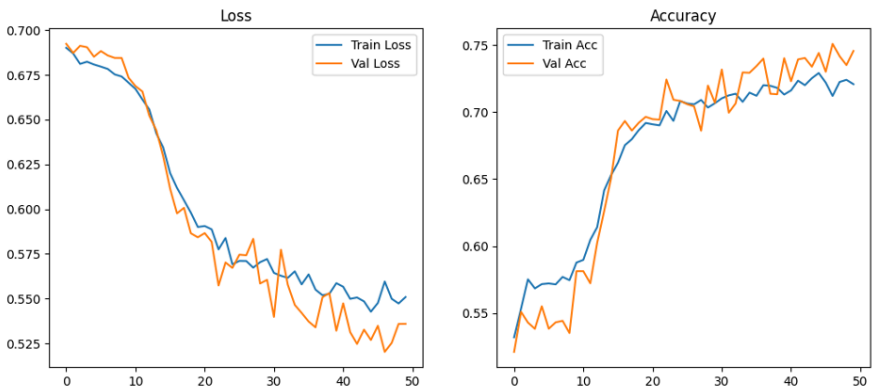


Figure 4: CNN - Model Accuracy & Loss (masking RNA and DNA)

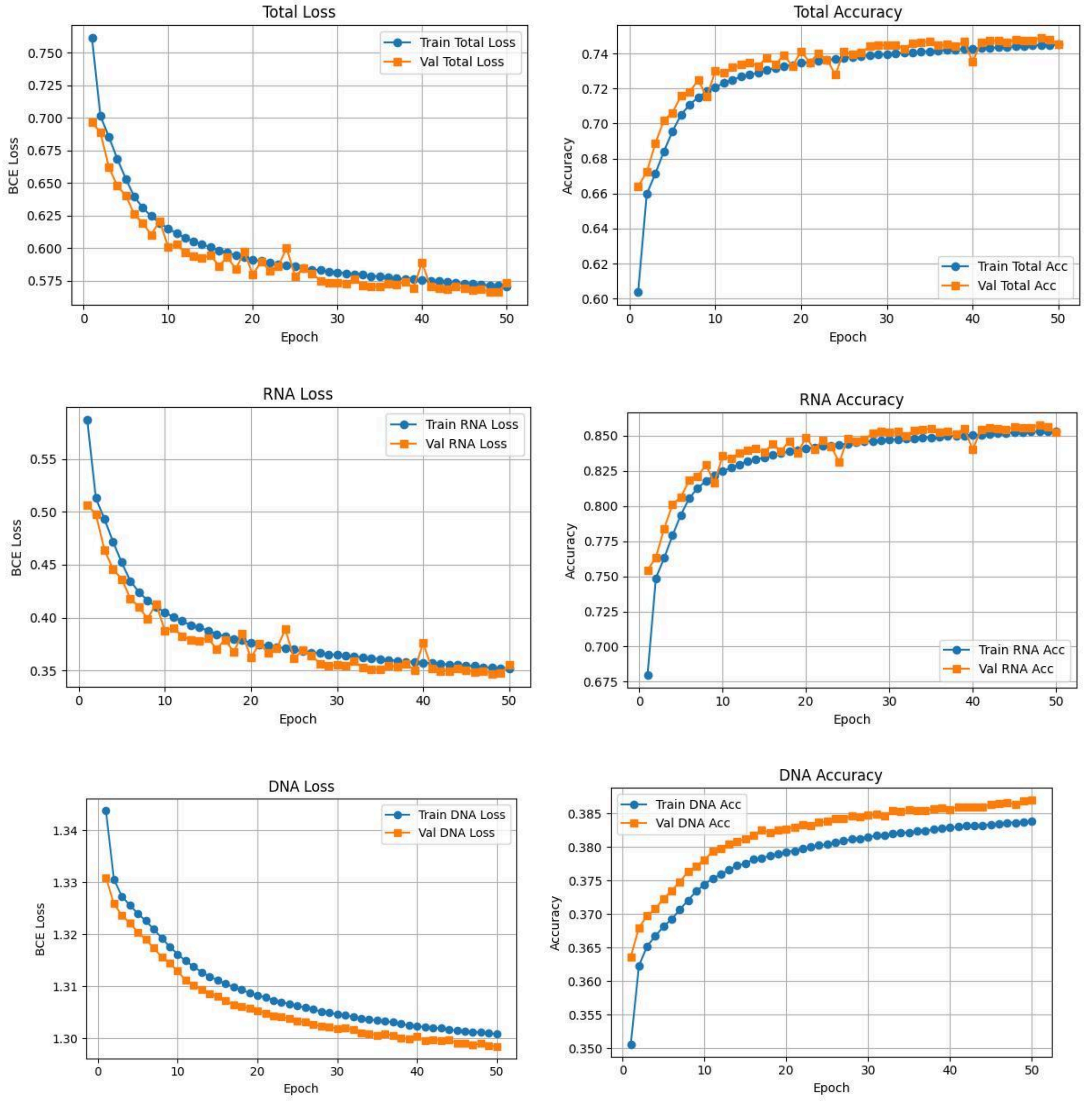



Figure 5: Transformer - Model Accuracy & Loss (masking RNA Only)

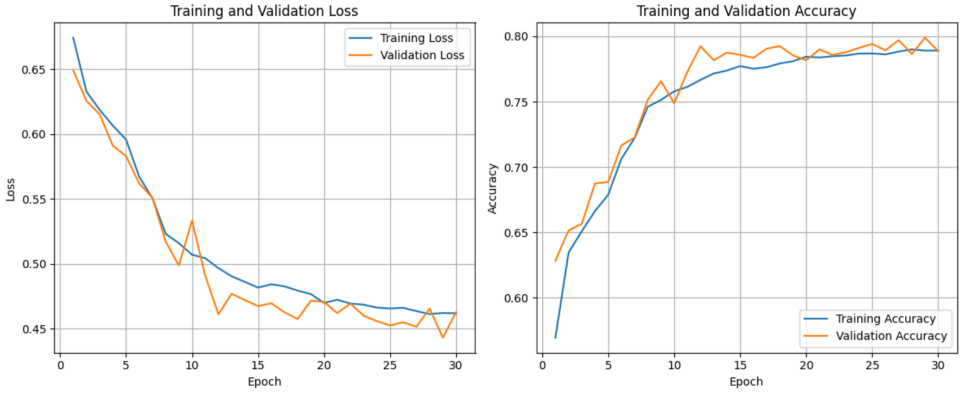


Figure 6: Transformer - Model Accuracy & Loss (masking RNA and DNA) - **20.000 data points, 50 epochs**

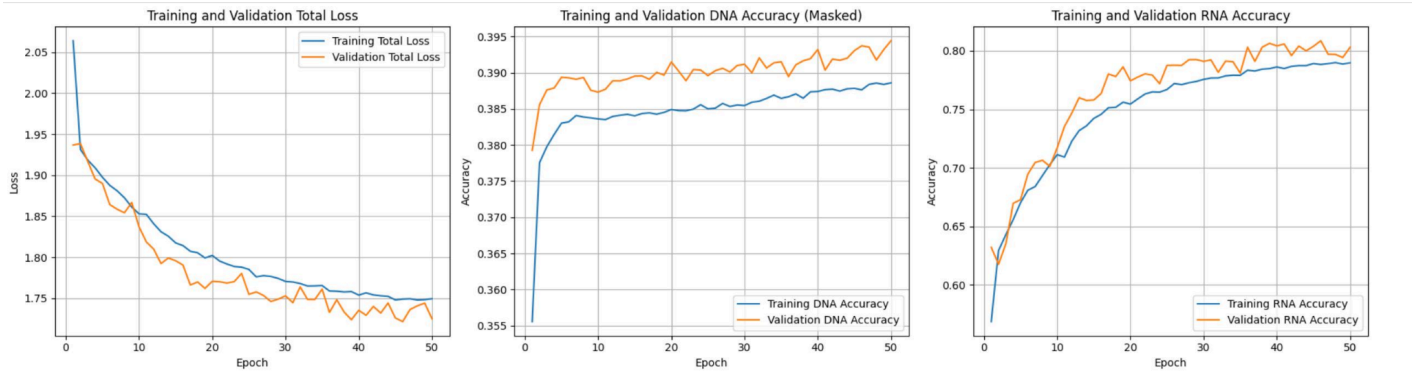



Figure 7: Transformer - Model Accuracy & Loss (masking RNA and DNA) - **100.000 data points, 9 epochs**

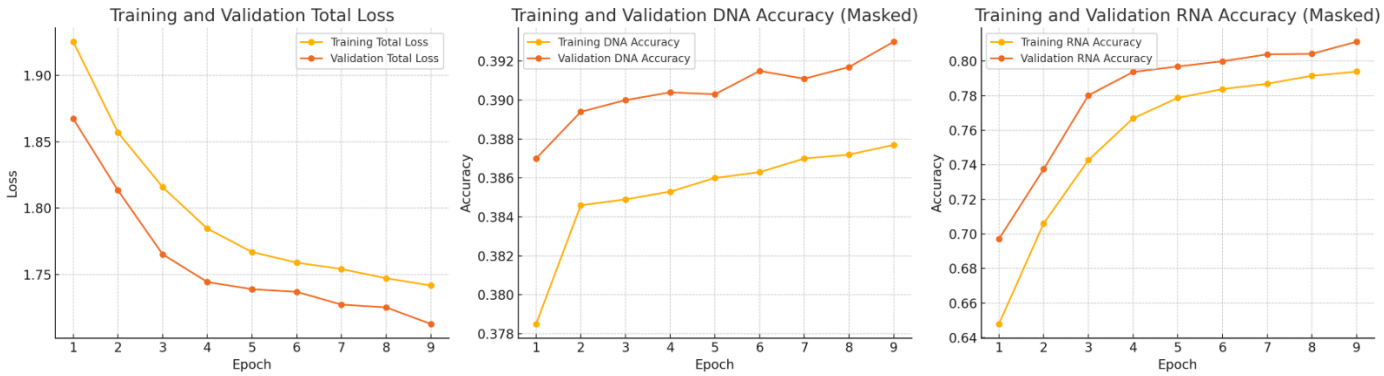



Figure 8: Sequence logo for whole-window (chrIX: 47,500–49,547 bp, results stored on the github) and (chrIV: 1,468,900–1,470,947 bp, results stored on the github)