Bacterial DNA in the Spotlight

May 1st, 2024

Manjot Nagyal
02-604 Fundamentals of Bioinformatics Project Final Report

Abstract

This study aimed to improve the prediction of *Escherichia coli* promoters using transfer learning with Nucleotide Transformer models. Pre-trained Nucleotide Transformer models with varying parameters and training datasets were fine-tuned using datasets from the Prokaryotic Promoter Database (PPD). Additional experiments included optimizing the accuracy of the models by varying the training steps and the LoRA (Low-Rank Adaptation) dropout rate. The models' performance was evaluated using a testing dataset comprising 865 natural and experimentally validated *E. coli* K-12 promoter sequences and 1000 randomly generated negative sequences. Results demonstrated that transfer learning with Nucleotide Transformer models significantly improved *E. coli* promoter prediction, with the best-performing model achieving an accuracy of 0.81.

Contents

1 Introduction

Bacteria play a crucial role in our understanding of human infections, developing diagnostics and vaccines, identifying antimicrobial targets, and designing drugs. Genomics can help in this understanding of what microbes do. Specifically, the study of prokaryotic promoters provides useful information on the regulated expression of genes. Bacterial promoters typically contain two short sequence elements at about −10 and −35 nucleotides upstream from the transcription start site, with consensus sequences TATAAT and TTGACA, respectively [1]. Although often conserved, most natural promoters that have been identified do not possess either of the consensus sequences fully conserved [1]. In fact, artificial promoters with these elements conserved tend to transcribe at lower frequencies than those with elements containing a few mismatches [1]. Therefore, accurately mapping natural promoter sequences is of great significance in microbial genomics.

By considering promoters as sequences analogous to language, the complex genetic patterns can be deciphered to identify specific regions of the genome using transformers. Transformers are neural networks capable of understanding relationships across long sequence lengths due to their ability to capture and employ content-aware gating (i.e., attention). In contrast, convolutional neural networks focus on element-wise interactions within a fixed receptive field [2].

Recent work has benchmarked the top bacterial promoter prediction tools, with the best three tools trained using machine learning techniques such as convolutional networks [3]. Moreover, with the advent of next-generation sequencing, there are a number of curated databases of prokaryotic promoters, such as the Prokaryotic Promoter Database (PPD) and RegulonDB, which focuses on *Escherichia coli K-12* promoters [4], [5]. Many of the prediction tools have also been trained on the RegulonDB database, despite these tools focusing on predicting a multitude of prokaryotic promoters [3].

In this work, we hypothesize that a transformer architecture will outperform convolutional neural networks in predicting *Escherichia coli* promoters. We employ transfer learning using pre-trained Nucleotide Transformer models with varying parameters and training datasets, including the NT 500m-1000g, NT 2.5b-1000g, and NT 2.5b-multispecies models. Initial experimentation included fine-tuning datasets derived from the PPD, which was used to identify the best pre-trained model to move forward with. Additional experiments improved upon this baseline fine-tuned model by fine-tuning with an *E. coli* subset of the PPD data. Our experiments include optimization of the LoRA (Low-Rank Adaptation) dropout rate, ranging from 0.1 to 0.9. We also compare the performance of models trained for 1000 and 10000 steps. The models' performance is evaluated using a testing dataset that was used in a previous benchmarking study to ensure a comparative analysis with other tools tested [3].

Thus, the goal of accurately and comprehensively mapping promoter sequences can help our understanding of bacterial gene regulation and, in turn, make an impact on human health and society.

2 Methodology

2.1 Model Training

2.1.1 Pre-training Nucleotide Transformer

The Nucleotide Transformer (NT) models are masked language models pre-trained on whole-genome DNA sequences. Developed by InstaDeep, NVIDIA, and TUM, these foundation models have demonstrated superior accuracy in human promoter prediction tasks compared to existing methods [6]. In this project, we employed the NT 500m-1000g, NT 2.5b-1000g, and NT 2.5b-multispecies models. The 500m-1000g is a relatively compact model with 500 million parameters, trained on 3,202 genetically diverse human genomes. Similarly, NT 2.5b-1000g is a larger model trained on the same genomes. The NT 2.5b-multispecies model, also comprising 2.5 billion parameters, was trained on 850 genomes from various species, including *E. coli* whole genomes.

2.1.2 Nucelotide Transformer Architecture

The hyperparameters were largely consistent across the models. The input sequences were tokenized into 6-mers. The models were trained using a warmup phase lasting 16000 updates, during which the learning rate increased linearly from 0.00005 to 0.0001. 95% of the data was used for training.

During training, 15% of the input tokens were randomly masked to encourage the model to learn meaningful representations from the surrounding context. 80% of the masked tokens were replaced with a special mask token, 10% (only for the 2.5B Multispecies model; otherwise 0% for the other models) were replaced with a random token, and the remainder were left unchanged. This masking strategy, inspired by the masked language modeling task in natural language processing, helps the model predict the original tokens based on the context provided by the unmasked tokens [2].

The models used 20 attention heads in their transformer layers. Attention heads are the mechanism in the transformer that weights the pairwise importance of different input elements in sequences based on the cosine similarity calculation. Essentially, a weighted average of the similarities are computed. Token-level dropout, where entire tokens are randomly removed, was applied during training. The 2.5B models had an embedding size and feed-forward network dimension of 2560 and 10240, respectively, and 32 transformer layers. The 500M model used smaller embeddings (1280) and feed-forward dimensions (5120) and fewer layers (24).

The embedding size refers to the dimensionality of the learned representation for each token in the input sequence. A larger embedding size allows the model to capture more complex patterns and relationships between the tokens. The feed-forward network dimension determines the size of the hidden layer in each transformer layer, which is responsible for processing and transforming the input representations. A larger feed-forward dimension increases the model's capacity to learn intricate patterns in the data. The 2.5B models also utilized 32 transformer layers, which are stacked on top of each other to enable the model to capture hierarchical representations of the input sequences. The increased depth of the model, combined with the larger embedding and feed-forward dimensions, allows the 2.5B models to learn more sophisticated patterns and relationships in the genomic data compared to the smaller 500M model.

This pre-training phase familiarizes the model with a wide range of genomic sequences, enabling it to comprehend different sequence patterns, contextual associations, and overall intricacies of genomic information. The resulting weight parameters encode these acquired understandings, which can be fine-tuned for the promoter prediction task.

2.1.3 Fine-tuning Nucelotide Transformer

Low-Rank Adaptation (LoRA) is an approach for fine-tuning large language models like the Nucleotide Transformer. LoRA freezes the pre-trained model weights, instead of tuning all the parameters of the pre-trained model which would cause overfitting. In this work, LoRA dropout hyperparameter was tuned from 0.1 to 0.9. This dropout probability determines the likelihood of a trainable parameter being randomly set to zero during training for a given batch. Dropout can be considered an optimization problem that prevents model overfitting [7]. Overall, LoRA has been shown to perform as well as full model fine-tuning while being more computationally and temporally efficient [8].

The positive dataset, or promoter dataset, was obtained from the PPD [4]. Figure 1 illustrates the composition and distribution of the PPD dataset used for fine-tuning. Notably, *E. coli* comprises only 6.7% of the total PPD. This subset of *E. coli* promoter sequences was used for a second round of fine-tuning, with models denoted accordingly throughout the paper. The rationale behind this second fine-tuning phase is that while the model's task is to classify *E. coli* promoters, it has been trained on promoters from various species. Fine-tuning again on the *E. coli* subset allows more of the model's capacity to be employed during the classification task on the test dataset.

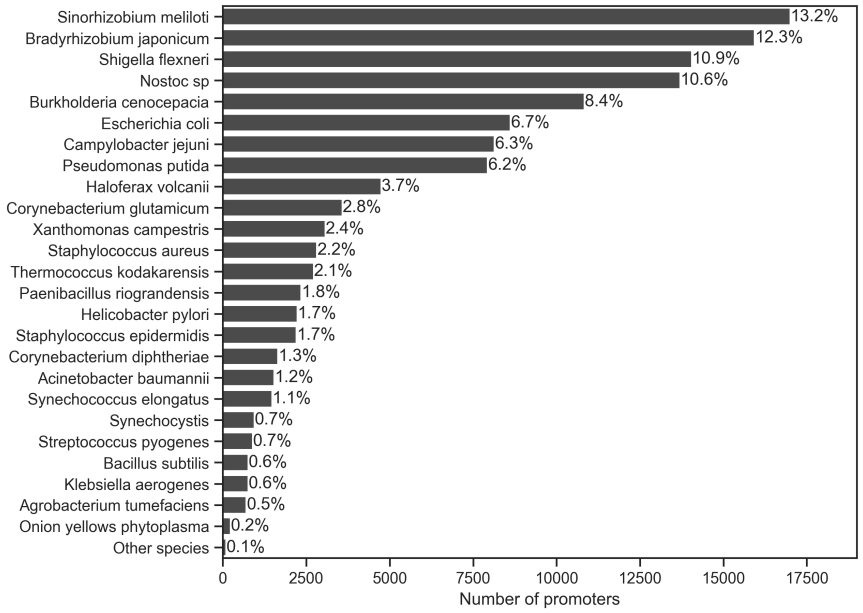


Figure 1: Distribution of species in the fine-tuning dataset.

The test dataset consisted of 865 natural and experimentally validated 81-base-pair promoter sequences from *E. coli* K-12 and 1,000 randomly generated sequences with nucleotide distributions similar to the positive sequences. This dataset, distinct from the PPD, was used to benchmark state-of-the-art promoter prediction tools [3].

2.1.4 Metrics of Assessment

This study considered four key metrics for the analysis of test outcomes: accuracy, Matthews Correlation Coefficient (MCC), sensitivity, and specificity. Accuracy served the dual purpose of evaluating the fine-tuning phase of the model.

Accuracy, denoted as the quotient of true positives and true negatives divided by the sum of all values, was employed to assess the model's overall effectiveness. MCC, a statistical measure used to reduce bias

stemming from imbalanced datasets, was computed. Given the higher prevalence of non-promoter sequences than promoter sequences within the test dataset, MCC served to mitigate the impact of this imbalance. Sensitivity, indicative of the likelihood of correctly identifying promoter sequences as such, and specificity, reflective of the likelihood of correctly identifying non-promoter sequences, were calculated to provide insights into the model's discriminatory capabilities.

2.2 Model Analysis

2.2.1 Overall Analysis

Platt calibration is a statistical technique that transforms the outputs of a classification model into a calibrated probability distribution across different classes [9]. This method refines the model's predictions, ensuring that the outputs reflect a true probability distribution. Calibration curves are utilized to evaluate the effectiveness of Platt calibration for probabilistic predictions made by binary classifiers [9]. These curves graphically depict the frequency of positive labels as a function of the predicted probabilities. More precisely, they represent an estimate of the conditional event probability, denoted as $P(Y = 1|\text{predicted probability})$, on the y-axis and the model-generated probabilities on the x-axis [10]. These model-generated probabilities arise from final layer embeddings being transformed by the nucleotide model. In this study, calibration curves played a critical role in determining the precision with which the transformer model's probabilistic outputs mirrored the actual class probabilities. This analysis assessed the model's capability to accurately classify sequences into promoter versus non-promoter classes.

2.2.2 Embeddings Analysis

In the context of the Nucleotide Transformer, analyzing the input and output embeddings can provide valuable insights into how the model learns to distinguish between promoter and non-promoter regions. By examining the embedding space, we can assess the model's ability to capture relevant patterns and relationships within the genomic sequences.

We analyzed the input and output embeddings produced by the Nucleotide Transformer. The input embeddings were constructed by mapping each nucleotide in an input sequence to a one-hot encoding vector.

To obtain the whole input, or pre-processed embeddings, we concatenated the vectors across all k-mers, resulting in a bag-of-words (k-mers) model. We examined this embedding to help understand how differentiable promoters and non-promoters are based solely on the presence and frequency of each k-mer.

The whole output, or post-processed embeddings, were constructed by averaging the outputs across the last m layers of the model. This embedding was the result of the transformer learning to add interaction information between k-mers to the original embeddings, capturing linear and nonlinear relationships between the k-mers and their positions in the genome sequence [11].

We expected the output embeddings to be substantially more separable than the input embeddings, as we anticipated non-promoter and promoter regions to share many k-mers but differ greatly in their joint distribution.

To visualize the embeddings, we used PaCMAP, a dimensionality reduction technique that attempts to preserve both local compactness and long-range distances in lower dimensions [12]. This results in a reduction that tends to maintain both shape and distance fairly well in lower dimensions.

2.2.3 Attention Maps Analysis

Attention mechanisms are a crucial component of transformer models, allowing them to focus on the most relevant parts of the input sequence when making predictions. By analyzing the attention maps generated by the Nucleotide Transformer, we can gain valuable insights into the model's decision-making process and identify the nucleotide positions that contribute most to the classification of promoter and non-promoter regions.

In this study, we employed salience mapping to analyze and visualize the importance of each nucleotide position in determining promoter regions in genomic sequences. Attention scores were assigned based on the predictive model's focus during classification tasks, indicating the significance of each nucleotide in classifying promoter versus non-promoter regions. Each sequence was represented as a k-mer, with attention scores calculated and normalized to account for sequence length and nucleotide frequency. The salience for each nucleotide was then computed by dividing the attention score by the frequency of the nucleotide, adjusted with a pseudo-count to avoid division by zero. This resulted in a map representing the relative importance of each position within the k-mer.

The salience maps were aggregated across four subsets of the test dataset: correctly classified promoters, misclassified promoters, correctly classified non-promoters, and misclassified non-promoters to compute an average salience map. The maps were visualized using logomaker [13].

The goal of this approach was to gain interpretability into the types of patterns the transformer model used to make its decisions. It can help us answer questions such as:

1. In aggregate, are certain regions of the sequence more important for decision-making than others?
2. What makes the model make a poor decision in regards to sequence patterning?
3. What nucleotide-position pairs are important for decision-making?

Not only can this analysis help us perform sanity checks on the underlying learned model, but it can also potentially help us learn new features of the underlying problem.

3 Results

3.1.1 All Models

The baseline established by benchmarking promoter prediction tools identified the top three prediction tools as CNNProm, iPro 70-FMWin, and 70ProPred, as identified in Figure 1 of [3]. For this study, initial experiments were conducted using the Nucleotide Transformer models: nucleotide-transformer-500m-1000g (NT 500m-1000g), nucleotide-transformer-2.5b-1000g, and nucleotide-transformer-2.5b-multi-species to compare the transformer's performance to the published baseline.

Although the validation accuracy was highest for nucleotide-transformer-2.5b-multi-species by the 1000th step of model training in Figure 2, the test accuracy was highest for NT 500m-1000g Figure 3. Compared to the benchmarked study, NT 500m-1000g outperformed two of the top three prediction tools. The specificity was higher for all transformer models compared to the benchmarked models, indicating that the transformer has a higher likelihood of correctly predicting non-promoters.

For further experiments and improvements in accuracy and other metrics, only the NT 500m-1000g model was considered.

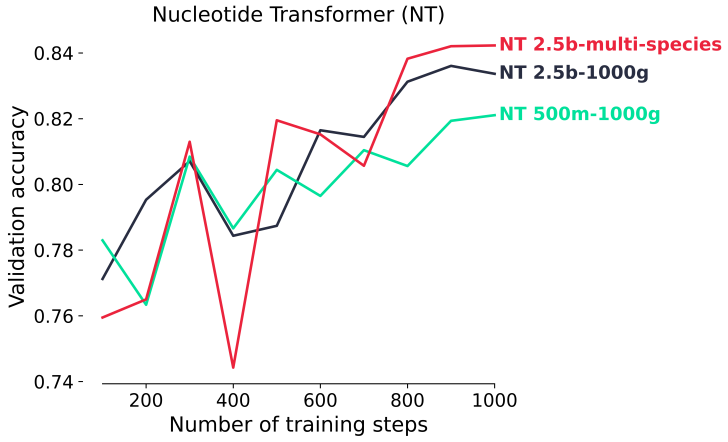
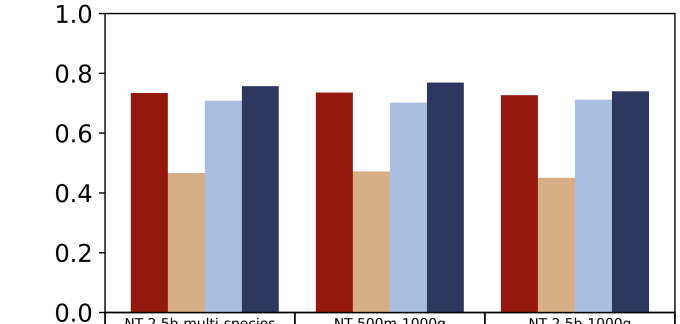


Figure 2: Validation accuracy of the three pre-trained Nucleotide Transformer models.



	NT 2.5b-multi-species	NT 500m-1000g	NT 2.5b-1000g
ACCURACY	0.73	0.74	0.73
MCC	0.47	0.47	0.45
SENSITIVITY	0.71	0.70	0.71
SPECIFICITY	0.76	0.77	0.74

Figure 3: Metrics used to evaluate performance of the three pre-trained Nucleotide Transformer models.

3.1.2 Nucleotide Transformer 500m-1000g

As identified in the initial experiments with the three pre-trained models, the small model with only 500 million parameters outperformed the larger models. To further improve the performance metrics of the model, three aspects of the fine-tuning process were considered: the LoRA dropout rate, the number of fine-tuning phases needed, and the number of steps during training.

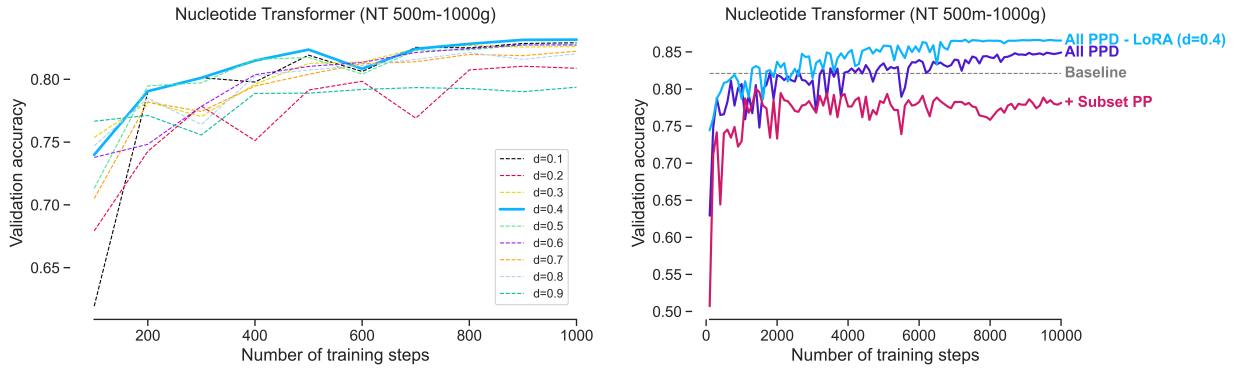
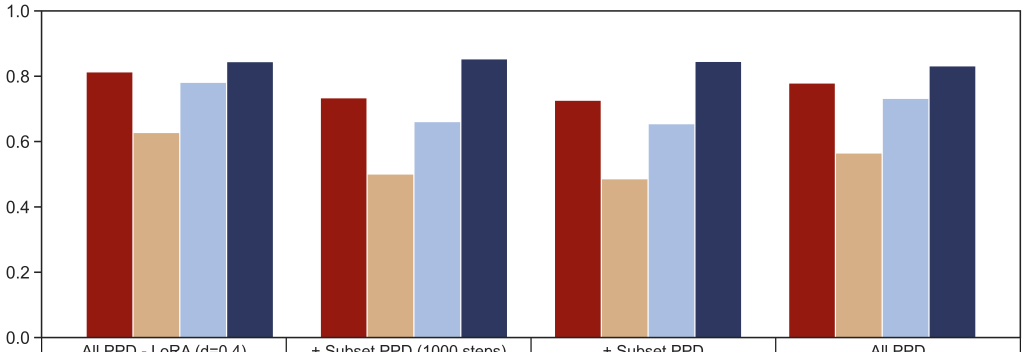


Figure 4: Validation accuracy of NT 500m-1000g model for a range of LoRA dropouts from 0.1 to 0.9 alongside validation accuracy plot with different fine-tuning datasets and LoRA dropouts (labeled on figure, otherwise d=0.01). Baseline corresponds to the highest validation accuracy from Figure 2, NT 2.5b-multispecies.



	All PPD - LoRA (d=0.4)	+ Subset PPD (1000 steps)	+ Subset PPD	All PPD
ACCURACY	0.81	0.73	0.73	0.78
MCC	0.63	0.50	0.49	0.57
SENSITIVITY	0.78	0.66	0.65	0.73
SPECIFICITY	0.84	0.85	0.85	0.83

Figure 5: Metrics used to evaluate the performance of different fine-tuning datasets, training steps, and LoRA dropouts (labeled on figure, otherwise 10,000 steps and d=0.01).

The LoRA dropout values ranged from 0.1 to 0.9, incrementing by 0.1, which was considered based on published work regarding reducing overfitting using LoRA dropout rates [7]. Although 5% of the PPD dataset was used for validation, the potential for overfitting arose from the fact that the PPD dataset contained only 3.5 million tokens, and the model could have simply memorized the dataset. Hence, hyperparameter tuning of the LoRA dropout was considered. The optimal LoRA dropout value was found to be 0.4, as seen in Figure 4, when trained for 1,000 steps. Furthermore, validation accuracy improved even more when the training was done for 10,000 steps. The default LoRA dropout rate was 0.1 and was used for all previous models. A LoRA dropout of 0.4 could indicate that the model was previously overfitting.

It is also worth noting that the test accuracy was highest for this model at 0.81, as seen in Figure 5, along with the sensitivity. This suggests that the model is more likely to correctly predict promoters compared to other models in Figure 3 and Figure 5.

Moving forward with model analysis, only the NT 500m-1000g All PPD - LoRA (d=0.4) model trained for 10,000 steps was considered.

3.2 Overall Analysis Results

To assess the model's performance in predicting promoter and non-promoter classes, we examined the calibration curve shown in Figure 6. The plot depicts the distribution of promoters and non-promoters across various predicted probabilities generated by the transformer model. The proximity of both classes to the perfectly calibrated line indicates effective model performance in predicting these two categories. However, the non-promoters deviate above the line, suggesting underestimation, while the promoters deviate below the line, suggesting overestimation. Despite these slight deviations, the calibration curve suggests that the model is well-tuned for the binary classification task.

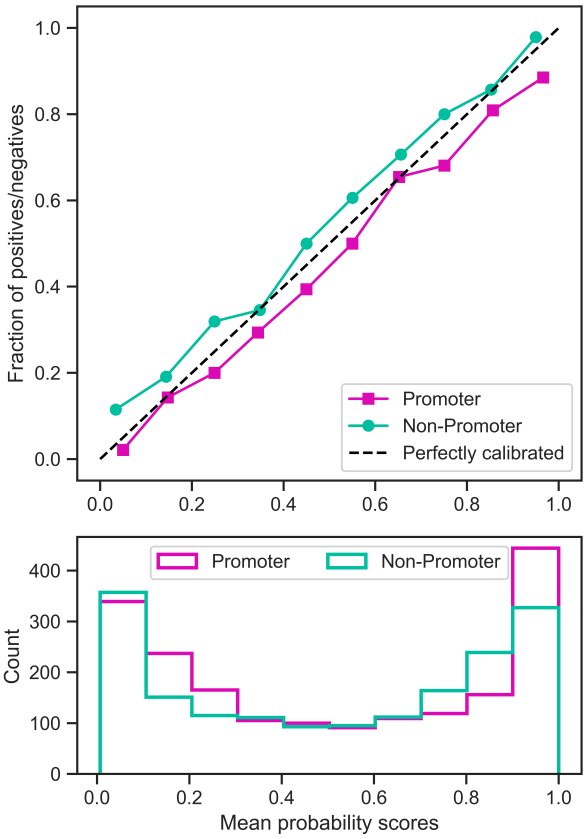


Figure 6: Calibration plots and distributions of probability scores for promoter vs. non-promoter classes.

3.3 Embeddings Analysis Results

To gain further insights into the model's decision-making process, we analyzed the input and output embeddings produced by the Nucleotide Transformer. We first evaluated the degree to which the embeddings can capture sequence information without applying the model (input embeddings), as shown in Figure 7, and then after applying the model (output embeddings), also in Figure 7. The PaCMAP projec-

tions of the embeddings reveal that the output embeddings of the model are substantially more separable than the input embeddings. This finding confirms that it is not the presence of specific k-mers that allows for distinction between promoters and non-promoters, but instead the presence, position, and frequency of k-mers in relation to each other. The transformer model effectively learns to capture these complex relationships, enabling better discrimination between the two classes.

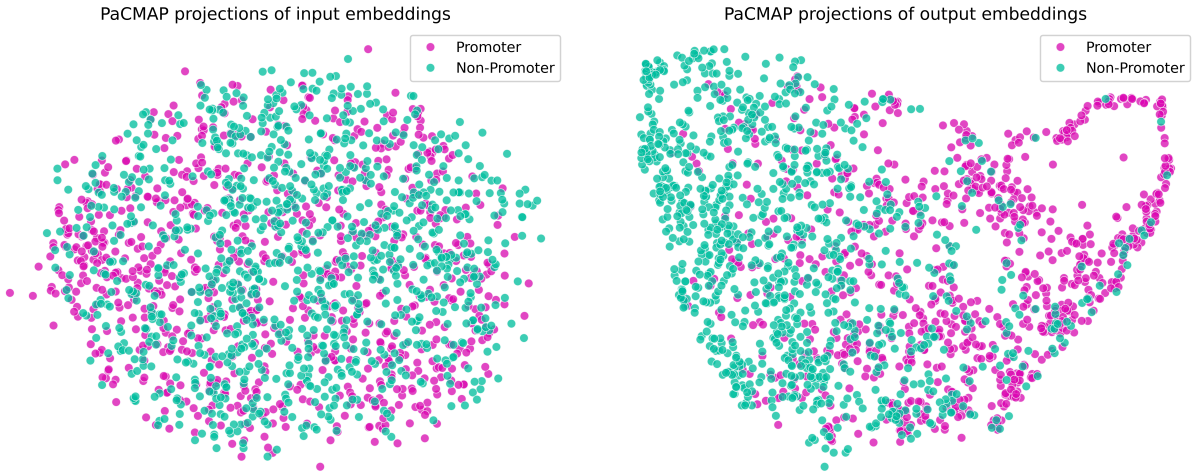


Figure 7: PaCMAP projections of embeddings shows the spatial distribution pre- and post-processing through nucleotide transformer. Each point represents an embedding showcasing the clustering and differentiation of classes.

3.4 Attention Weights Analysis Results

To further investigate the model's decision-making process, we conducted an attention weights analysis. Figure 8 illustrates the relative significance of each nucleotide within the sequence regions, with the vertical dimension of each nucleotide reflecting its importance in defining these regions. The promoter sequences, both the correctly classified and misclassified, exhibit distinctive nucleotide compositions near position 0, the transcription start site, distinct from those observed in non-promoter sequences. This observation suggests that the model has learned to focus on biologically relevant regions when making predictions.

The salience maps for misclassified promoters are challenging to interpret due to their complex visual patterns, whereas the maps for the other categories display clearer visualizations. Although these three maps provide distinct visual information, they vary from one another in terms of nucleotide salience at each position, indicating differential contributions to the predictive model's decision-making process. These variations in salience patterns across the different subsets of the test data provide valuable insights into the factors influencing the model's predictions and highlight potential areas for further improvement.

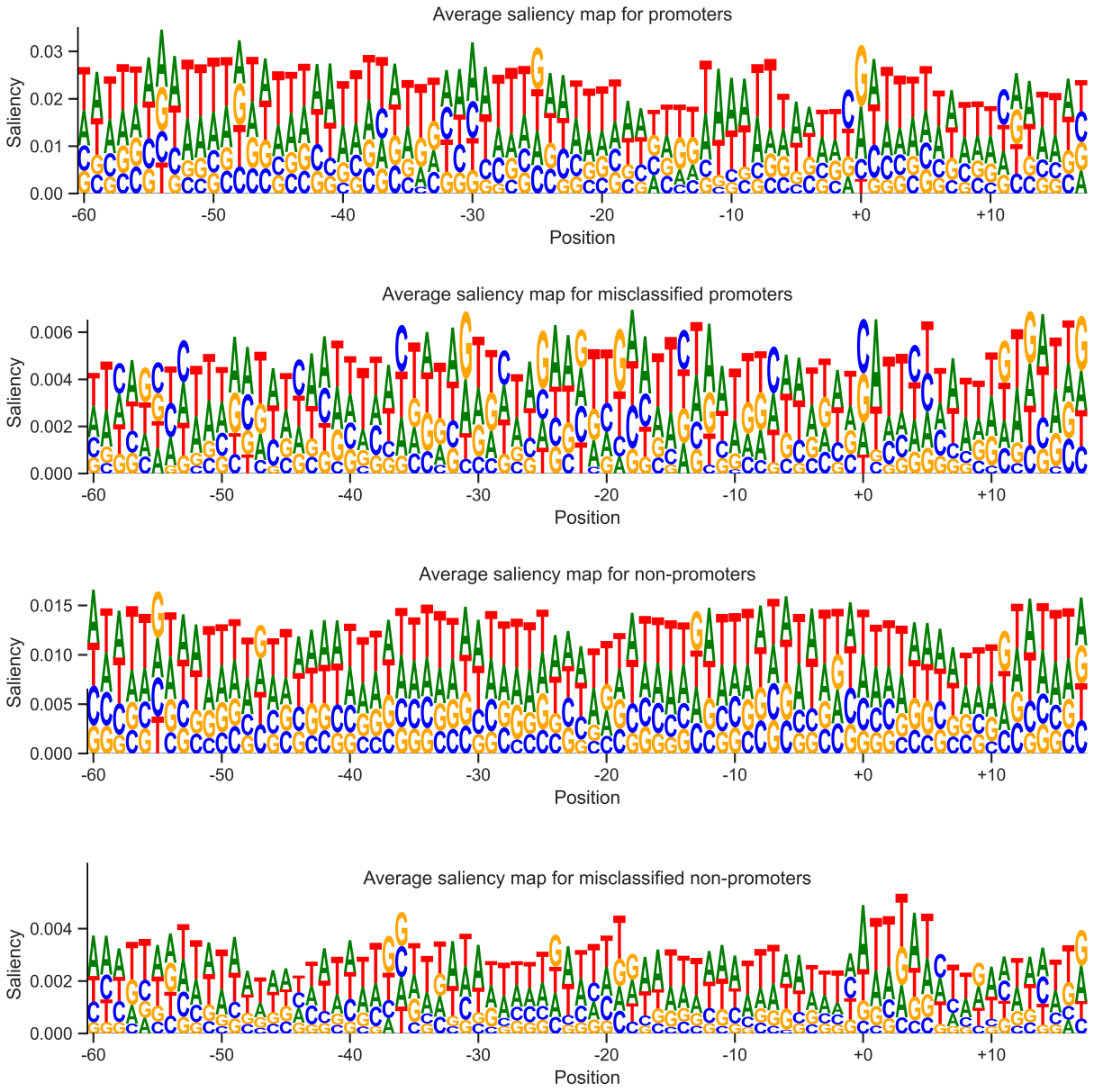


Figure 8: Average salience values for nucleotides across sequence length for the four test data subsets.

4 Conclusion

In this study, we demonstrated the effectiveness of transfer learning using Nucleotide Transformer models for improving the prediction of *Escherichia coli* promoters. By fine-tuning pre-trained models and optimizing the process, we achieved an accuracy of 0.81, outperforming previous benchmarks. Analysis of the model's performance through calibration curves, embedding visualizations, and attention weights provided insights into its decision-making process, confirming its ability to capture complex relationships between k-mers and focus on biologically relevant regions.

Future directions for this research include exploring other transformer architectures, focusing on bacterial DNA for unsupervised pre-training, utilizing curated *E. coli* promoter databases for fine-tuning, and investigating prediction performance for other bacterial species.

In conclusion, this study highlights the successful application of transfer learning with Nucleotide Transformer models for improving *Escherichia coli* promoter prediction, setting the stage for further advancements in microbial genomics and the deciphering of complex genetic patterns in bacterial genomes.

Bibliography

[1] J. Wright, *Gene Control*. Scientific e-Resources, 2019.

[2] M. E. Consens *et al.*, "To Transformers and Beyond: Large Language Models for the Genome," *arXiv preprint arXiv:2311.07621*, 2023.

[3] M. H. A. Cassiano and R. Silva-Rocha, "Benchmarking bacterial promoter prediction tools: Potentialities and limitations," *Msystems*, vol. 5, no. 4, pp. 10–1128, 2020.

[4] W. Su *et al.*, "PPD: a manually curated database for experimentally verified prokaryotic promoters," *Journal of Molecular Biology*, vol. 433, no. 11, p. 166860–166861, 2021.

[5] V. H. Tierrafria *et al.*, "RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12," *Microbial genomics*, vol. 8, no. 5, p. 833–834, 2022.

[6] H. Dalla-Torre *et al.*, "The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics," *bioRxiv*, pp. 2023–2021, 2023.

[7] Y. Lin *et al.*, "LoRA Dropout as a Sparsity Regularizer for Overfitting Control," *arXiv preprint arXiv:2404.09610*, 2024.

[8] E. J. Hu *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[9] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 625–632.

[10] D. S. Wilks, "On the combination of forecast probabilities for consecutive precipitation periods," *Weather and forecasting*, vol. 5, no. 4, pp. 640–650, 1990.

[11] J. Clauwaert, G. Menschaert, and W. Waegeman, "Explainability in transformer models for functional genomics," *Briefings in bioinformatics*, vol. 22, no. 5, p. bbab60, 2021.

[12] H. Huang, Y. Wang, C. Rudin, and E. P. Browne, "Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization," *Communications biology*, vol. 5, no. 1, p. 719–720, 2022.

[13] A. Tareen and J. B. Kinney, "Logomaker: beautiful sequence logos in Python," *Bioinformatics*, vol. 36, no. 7, pp. 2272–2274, 2020.