Mining Molecular Logics through Human Language: Predicting and Decoding Transcription Factor Logics on Gene Expression through LLM and transformer

Stanford CS224N Custom Project

Gyu (Gyuhyeon) Kim
Department of Biochemistry
School of Medicine
Department of Computer Science
Stanford University
gyukim@stanford.edu

Abstract

Gene expression is an essential step of biological processes, and there has been growing interest in predicting its activity from biological markers. Transcription factors are key modulators of this process, yet their capacity to predict gene expression has remained elusive. At first glance, human language and gene expression may seem fundamentally disconnected. However, the project presented the first case of predicting gene expression level through transcription factors using two different approaches: (1) encoding molecular properties into numeric values and using a custom transformer model and (2) encoding molecular properties into human language and using pre-trained human Large Language Model (LLM) for training. Surprisingly, both approaches showed similar or better performances compared to the baseline. Further analysis identified similar limitations of the two models and found out interesting patterns and biological interaction networks from self-attention analysis, which identified a known interaction and potentially provided promising leads to decode transcription factor logics on gene expression.

1 Key Information to include

- Mentor: Kaylee Burns
- External Collaborators (if you have any): No, Sharing project: No

2 Introduction

Can human LLM be used to decode unknown molecular properties? Gene expression is a core process of cellular activities and is regulated by multiple layers of biological mechanisms. Transcription factor is a major class of proteins known to directly regulate gene expression. [1] However, we currently have minimal understanding of the impact and capacity of transcription factors on predicting gene expression because of the large number of transcription factors co-occurring at the same time. The goal of this project is to accurately predict gene expression from transcription factor profiles by two different approaches and further extrapolate some meaningful molecular syntax between those factors. The first approach is directly encoding biological information into a custom transformer model, and the second approach is encoding biological information into a human language format and leveraging human LLM for the training. While applying transformer architecture for solving biological problems has shown some successful results from recent literature [2, 3], encoding molecular information into human language format and directly training on human LLM has never been reported before. It's odd to expect human LLM to be useful in predicting unknown molecular properties. Surprisingly,

the result showed that both approaches can successfully model transcription activity with similar or better performance levels compared to the baseline. This project provides an important initial effort to predict gene expression in a term of transcription factors. Additionally, it provides an interesting and exciting potential for applications of human LLM beyond the language contexts.

3 Related Work

3.1 Large Language Model in Genetics and Molecular Biology

Application of LLM in genetics and molecular biology is an actively growing field. Many biological properties are represented as a sequence, such as base pair sequences of DNA and RNA or amino acid sequences of proteins, and this sequential information implies natural properties of biological functions. Thus, several efforts have been made to build custom LLM focused on a certain biological property using BERT and GPT architecture. [4] For example, DNABERT, LLM on DNA, was made by pre-training on DNA from various organisms and utilized for various DNA-related tasks, such as predicting mutation effect on DNA or predicting protein bindings, through supervised fine-tuning. [5] However, the application of human LLM in genetics or molecular biology has been mostly limited to mining information for gathering information from literature for drug discovery or repurposing based on existing publications. [6] Encoding molecular information into human language and training them on human LLM to predict molecular properties is the novel approach taken in this project, which motivated to test how much human LLM can be directly useful for modeling molecular properties.

3.2 Machine learning approach on predicting gene expression

There have been a few studies of predicting gene expression from biological markers, especially from epigenomic histone markers which is another biological modulator for gene expression. Machine learning has shown promising results in modeling the gene expression level using the histone markers and providing initial evidence of decoding "histone code". A recent study showed a successful prediction based on a transformer-based model. [2] Before this study, the existing models were based on deep learning architectures like convolutional neural networks (CNN) or recurrent neural networks (RNN). [7, 8] However, these architectures have intrinsic weaknesses in that they only take local information and can't effectively model the distant dependencies while it has been known that distal interactions are important for gene regulation. The recent study tried to model the gene expression level using the histone marker through a transformer (1) to properly incorporate distant interactions among markers, (2) to improve the prediction power, and (3) to extract meaningful biological insights through transformer modeling, mostly based on self-attention values, which couldn't be captured by other type of architectures. This transformer-based model (Chromoformer) achieved the highest performance on this task compared to other existing models using the same input, which highlighted the strength of the transformer on gene expression prediction. [2] Collectively, these preceding works provide a reliable baseline metric for model performances and intuition on feature engineering and encoding methods of molecular marker information into numeric representation for this project.

4 Approach

In this project, two approaches were considered. (1) Using the custom-built transformer model (Approach I, Model I) and (2) using a pre-trained LLM model for regression (Approach II, Model II). In the first custom-built transformer model approach, the architecture was designed to take a binary representation of the transcription factor binding profiles as input and predict single float value output (gene expression level) similar to the previous study. The transcription factor binding profile has been processed into a binary matrix, and each position is sequentially fed into encoder units where dot product self-attentions are calculated between units. To integrate all the information from encoders, one decoder unit has been placed and fed with EOS. The output from the decoder is provided to the subsequent fully connected neural network with one hidden layer followed by the output layer with the ReLu activation function, which predicts one float value. (Figure 1).

For the second LLM-based approach, the original molecular binding data was transformed into a sentence format of human language. (Table 1). Then, these sentences were tokenized and used as

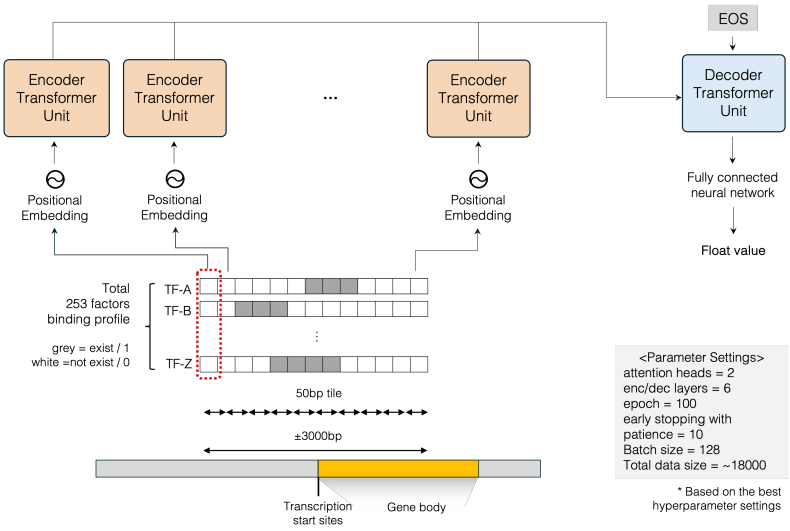

Figure 1: Model Architecture of custom transformer model for Approach I for a single input

input for supervised fine-tuning of DistilledBERT model for the regression tasks of predicting the transcription activity. To use DistilledBERT model for the regression task, the last hidden states were fed into a linear transformation layer to predict a single float value (gene expression level) instead of performing SoftMax in the original structure.

	Approach I	**Approach II**
Input Description	binary matrix of binned binding profile	human language sentence format of binding information
One Example Input	(253, 120) numpy array	"The gene has a set of transcription factor A, B, and Z"
Architecture	custom transformer model for regression	distilledBERT model for regression

Table 1: Table of comparison between Approach I and Approach II

For the baseline, Pearson R values from the most relevant study that performed gene expression level prediction using other markers were used. [2] The first approach of using transformer to predict biological properties from sequential binding profiles has been adopted from a few previous studies in a relevant context. The second approach of changing molecular data into human language format and fine-tuning based on the pre-trained human LLM is the novel approach to test if a human large language model can be used to understand molecular information simply formatted in human language format. For the architecture and coding, the majority of the scripts for data pre-processing, training, and downstream analysis were self-written with some assistance from ChatGPT. [9] The original PyTorch transformer unit was directly used for Approach I [10], and part of the CS224N HuggingFace Tutorial script was used for Approach II. [11] (Annotated in coding submission)

5 Experiments

5.1 Data

For the input data, 253 transcription factor binding profiles data was obtained from ReMap repository where they provide a quality-controlled dataset. [12] For approach I, the data was further processed for assigning specific genes and binning the binding information. Thus, a single input is represented as 253 x 120 2D numpy binary matrix, which represents 253 transcription factors binding information

across 120 bins for each gene. For the input data for Approach II, from the same ReMap data, a set of transcription factors were obtained from each gene and formatted as a human language sentence in a uniform way starting with "The gene has a set of transcription factors...". When a gene contains only one transcription factor, the sentence was phrased as "The gene has a transcription factor ...". The sentence was tokenized by distilledBERTFast tokenizer and used as an input with proper padding. For the target data (y values) representing transcription activity, the experimental data of measuring transcription activity, ChIP-exo, was obtained from the public repository and processed through a bioinformatics pipeline. [13] The collected values were log-transformed considering the wide ranges of the values. The target data was commonly used for both Approach I and II. The total dataset consists of around 18000 genes.

5.2 Evaluation method

Pearson R value, between true target values and predicted target values, is the standard metric reported from other studies done similar tasks. [2] [7] In literature, the range of Pearson R values after 3-4 cross-fold validations are reported to compare the model performances. For training and hyperparameter tuning, Mean Squared Error (MSE) was used as a loss function, but the final model performance comparison was reported based on Pearson R value to match with the literature baseline.

5.3 Experimental details

5.3.1 Approach I. Custom-built transformer model

As an experiment to find the best-performing model, different settings of parameters were tested. **(1)** In a fixed initial learning rate (0.0001), different batch sizes were tested (32, 64, 128), and the bigger batch size resulted in lower validation MSE. **(2)** With a batch size of 128, a lower learning rate of 0.00001 was tested to decrease the loss fluctuation during the training and further reduced validation MSE. **(3)** The original settings were to run all 100 epochs, but to prevent overfitting, early stopping with 10 epoch patience was implemented and reduced MSE. **(4)** Considering the imbalance of the target value distribution (and poor prediction accuracy on target values with lower occurrence rates), weighted loss functions based on the data frequency and oversampling method were tested separately, and oversampling resulted in better performances in MSE values to overcome the imbalanced dataset issues. **(5)** On top of the aforementioned determined settings, 54 different combinations of hyperparameter settings were tested to select the final model. The combinations include the initial input neural network dimension (64, 128, 256), number of attention heads (2, 4, 8), number of encoder/decoder layers (3, 6), and the hidden dimension of the fully connected neural network at the end (64, 128, 256). The combination of (256, 2, 6, 128) resulted in the lowest validation error (MSE: 0.5052). Overall, the experiments achieved a significant reduction in validation MSE loss from the initial model (from 0.8745 to 0.5052). After the experiment, the final configuration was determined as follows. **Final configuration**: learning rate: 1e-5, batch sizes: 128, initial input neural network dimension: 256, number of attention heads: 2, number of encoder/decoder layers: 6, the hidden dimension of the last neural network: 128, Epoch: 100, Optimizer: Adam, Loss: MSE, Early stopping with a validation set, oversampling for low occurrence values. (Detailed results available in the Appendix). Running 1 epoch roughly took around 15 seconds in T4 GPU.

5.3.2 Approach II. Fine-tuning of pre-trained LLM model

The initial training was set with a learning rate: 1e-4, batch size: 32, epoch: 3. (1) The batch size of 64 was tested, but it resulted in an error due to the lack of memory (over Google Colab Pro T4 GPU). (2) Different learning rate 1e-5 was tested, but it didn't improve test MSE loss. As mentioned in Approach I, target values have an imbalanced distribution, and the prediction was observed to be more accurate on data points with originally high target values. (3) To overcome this issue, two different approaches were tested, (a) Normalized Root Mean Square Error (NRMSE) loss function to reflect the scale of target values and (b) oversampling low occurring entries. NRMSE was tested in different learning rates (1e-4, 1e-5), but neither showed improvement on test MSE loss. Oversampling was implemented with different learning rates (1e-4, 1e-5), and the learning rate with 1e-4 exhibited a decent reduction in test MSE loss (0.84) and an increase in Pearson R value (0.74) compared to the original setting without oversampling (MSE: 0.87, Pearson R: 0.58). (4) To further improve the model, the AdamW optimizer was tested with a learning rate scheduler with different initial learning rates (1e-4, 5e-5, 1e-5) with and without oversampling. Though the model consistently

performed better with oversampling, still the model trained with fixed learning rate (1e-4) and Adam optimizer performed better. After the experiment, the final configuration was determined as follows. **Final configuration**: learning rate: 1e-4, batch sizes: 32, Epoch: 3, Optimizer: Adam, Loss: MSE, oversampling for low occurrence values. (Detailed results available in the Appendix). Running 1 epoch roughly took 5 minutes in T4 GPU.

5.4 Results

Annotation	Model Name	Test Pearson R	Test MSE	Source
Baseline	AttentiveChrom-reg	0.82 - 0.83	-	[2]
	DeepChrom-reg	0.83 - 0.84	-	[2]
	HM-CRNN-reg	0.82 - 0.83	-	[2]
	Chromoformer-reg	0.84 - 0.86	-	[2]
Approach I	Custom Transformer model	0.85 - 0.86	0.56 - 0.59	This Project
Approach II	LLM-based model	0.85 - 0.88	0.74 - 0.91	This Project

Table 2: Comparison between the Baseline and the tested models

For the baseline, the Pearson R values reported from the literature that conducted a similar task were used. (Table 2) [2] Test Pearson R values were collected from different subsampling with the same model and reported as a range. The literature reported multiple Pearson R values, but the above values are from the best-reported performance from each model.

Both Approach I and II surprisingly achieved high Pearson R values compared to the baseline values and slightly better in the case of Approach II. This was an unexpectedly high performance considering the baseline model was designed to include more complex interaction information and wider regions of genomic contexts compared to the models made in this project. This might have been driven by the differences in the input data set. All the baseline models used approximately a dozen of epigenetic markers to predict transcription activity while 253 transcription factors were used in this study. Thus, it could be the case that a larger number or the information of the transcription factor binding profile itself contains more useful information to predict transcription activity.

Particularly, the expected performance for Approach II was very low because human LLM is not considered to be relevant for understanding or predicting molecular properties like gene expression. However, it achieved remarkably high Pearson R values comparable to or higher than the baseline. There are a few hypotheses about its high performance. (1) The complex settings of the training model (6 encoder layers and 12 attention heads) with many inputs (approximately 18000) could have enabled extracting some patterns to manage the regression task. Additionally, (2) since biological events are strictly regulated by molecular rules and patterns, learning those patterns during the fine-tuning process could be sufficient to train the regression model even though the molecular information itself doesn't exist in human LLM.

The overall result shows that (1) Transcription factor binding information can successfully predict gene expression level using a transformer, and (2) Human LLM successfully predicted gene expression level from human language representation of molecular information

6 Analysis

6.1 Analysis comment on prediction tasks and the case of failures

The initial visualization of scatter plots between predicted and true target values revealed a few important limitations of the models. (Figure 2)

(1) First, it shows extremely poor prediction on data entries with zero transcription factor binding. An example of a scatter plot between prediction and true values (Figure) shows a thin line around -8. Those are all the entries with zero transcription factor binding (zero binding). In that vertical line, the true values are distributed across a wide range, which could be potentially explained by the effect from other transcription factors not included in the dataset or non-transcription factor-related biological effects. However, the prediction for zero bindings is all uniformly at around -8 because the inputs are

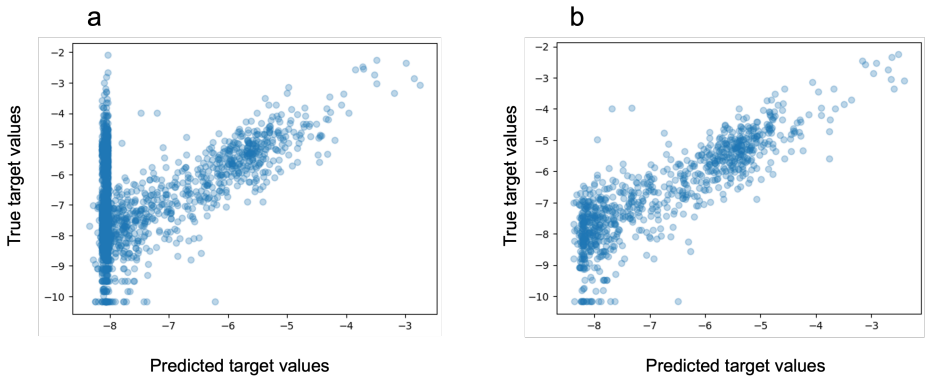

Figure 2: Example of scatter plot with predicted target values on the x-axis and true target values on the y-axis. From Approach I model. (a) With zero binding data. A thick horizontal line is from data input with zero transcription factor binding. (b) Without zero binding data.

the same for those entries. This issue driven from zero binding was common for both Approach I and II. Thus, those data points were later removed during the training and testing processes for accurate training and interpretation of the results. (Figure b) (2) Second, the model shows worse performances on lower target values. As shown in the Figure, the model could accurately predict as low as -8 while the true values ranged further down to -10, which suggests the model might have a lower threshold for the prediction. These issues were also observed in both Approach I and II and might have been caused by the low abundance of samples of those ranges of low values. Despite the implementation of oversampling on lowly abundant target values, this lower threshold issue still remains. Thus, other methods need to be considered for the further improvement of this model.

6.2 Visualization of attention values

To further analyze the meaningful interaction between transcription factors used for prediction, encoder self-attention values were visualized from both Approach I and II. (Figure 3)

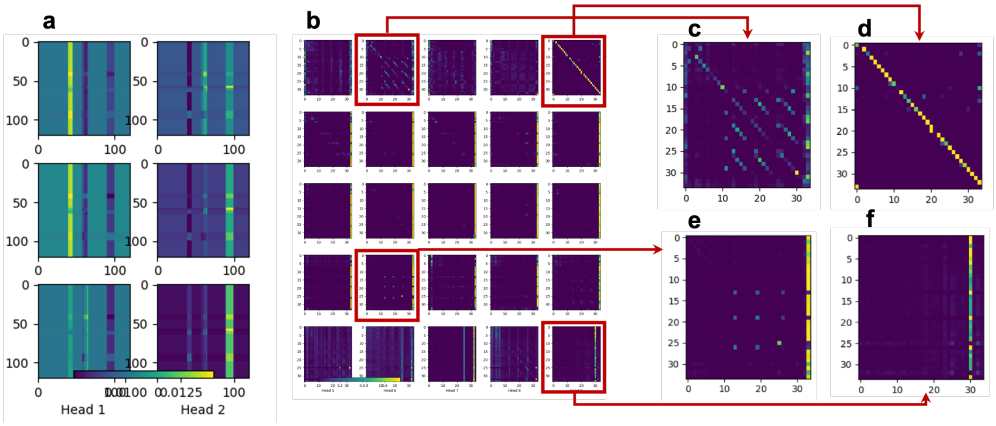

Figure 3: Example of encoder self-attention heatmaps. (a) Approach I. Part of the encoder heatmap visualization from three encoding layers and two attention heads. (b) Approach II. Part of the encoder heatmap visualization from five different encoding layers and five different attention heads. (c) - (f). Different patterns were observed. (c) scattered off-diagonal pattern. (d) diagonal pattern. (e) grid pattern. (d) horizontal pattern

The attention heatmap from Approach I mostly showed a mix of horizontal line or rectangular patterns while the patterns from Approach II were more diverse. Four commonly observed patterns of high

self-attention values were highlighted in the figure: a few consecutive off-diagonal positions are highlighted which will be called a scattered off-diagonal pattern (Figure c), highlighted on diagonal positions (Figure d), highlighted with a grid pattern (Figure e), highlighted as horizontal lines (Figure f). The most intriguing pattern was a scattered off-diagonal pattern only observed from the Approach II model. This is because, in Approach II, each transcription factors are represented as two to four tokens after the tokenization, such as "CEBPB" turns into 'ce', ' b', 'p', 'b', and "PBX2" turns into 'p', 'b', 'x', '2' after tokenization. Thus, the high self-attention between different transcription factors is expected to show up as a partial off-diagonal pattern. In the Approach I model, the high self-attention between transcription factors is expected to show up as rectangular or block of lines because each transcription factor binding is represented as a couple of bins, and the interaction between bins are rectangular shape (or straight blocks of lines) as shown up in Figure 3.

6.3 Comparative analysis of attention values between models

Both Approach I and II showed a similar range of Pearson R values on the regression tasks against the same target values. To analyze if two models are making the predictions through a similar self-attention mechanism, the off-diagonal positions of high attention values (top 0.1% among all attention values of each input) were collected from each model and plotted as a histogram to compare the high-level attention landscape. (top 0.1% was an arbitrary cutoff to minimize the false positive cases) (Figure 4)

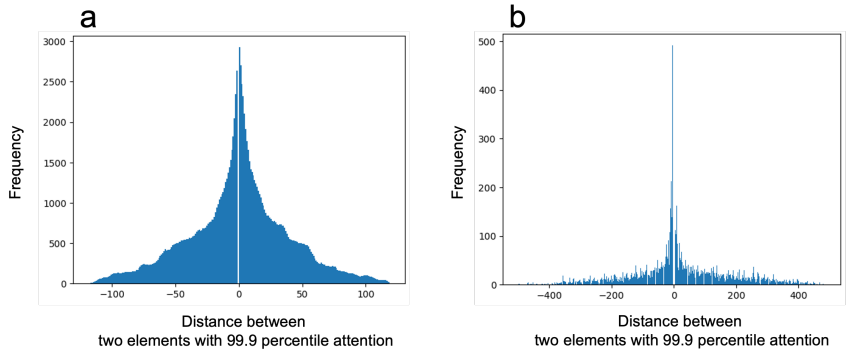

Figure 4: The histogram of the distance between two elements with high attention values (top 0.1% among all attention values in each input) from two models. (a) From Approach I model. (b) From Approach II model. The distance was calculated as "row index - column index" at the position with a high attention value.

The histograms commonly show higher frequency at shorter distances, which suggests high attention weights are more frequently observed between tokens closer to each other. The histogram from Approach II showed an extreme frequency at a short distance and a visually steeper decrease of frequency as the distance gets longer compared to the one from Approach I, but the overall frequency and symmetric patterns looked similar between the two models. This result briefly suggests that the self-attention landscape between elements (tokens) between the two models is comparable to each other.

Additionally, pairs of important interactions between transcription factors were extracted based on a high attention weight (Top 0.1% of all attention weights for each input) and visualized as a graph network. (Figure 5) Model I resulted in more factors in a core region (Inside of the red circle. Visually determined.) where dense connections are observed. Interestingly, all the factors observed in the core region of Model II were also detected in the core region of Model I except one (EGR1), which suggests that both models have something in common even though how the information was given and trained was vastly different. Interestingly, an interaction between MYC and MAX was detected in the core regions of both models, which is a well-renowned interaction pair of regulating gene expression. This provides promising evidence that the analysis based on self-attention can detect some real biological interactions. More systemic analysis and validation are to be explored on this result in the future.

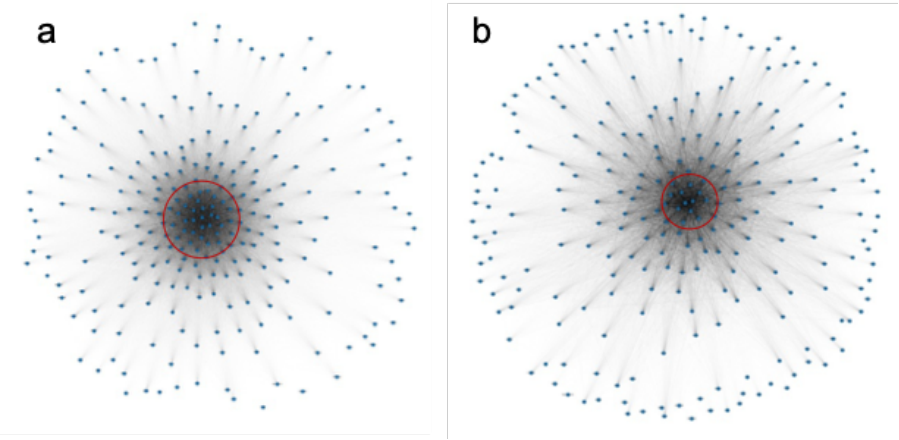

Figure 5: The graph networks for transcription factor interaction based on self-attention weight (top 0.1%) (a) From Approach I model. (b) From Approach II model. The core region is marked as a red circle. A Single blue dot indicates each transcription factor. The thickness of a line indicates the frequency of interactions between two nodes.

7 Conclusion

The project successfully modeled gene expression from transcription factors for the first time and achieved similar or better performances than the baseline using two different approaches. Particularly, the second approach of predicting molecular biology properties through human language encoding and LLM was a novel method and unexpectedly showed high performance. Remarkably, both models showed promising preliminary results that could capture biologically meaningful interactions through self-attention analysis. As a primary limitation of this project, a prediction task of biological properties could be varied depending on cellular contexts. Here, a dataset from only one model cell type, K562, has been tested. Future work of testing a few more model cell types using the same approach will provide more comprehensive view of the model performances. Additionally, for extracting meaningful biological interactions, it may require more in-depth self-attention analysis and cross-validation with other literature.

8 Ethics Statement

The second approach in this study of formulating molecular biology into human language and using human LLM showed a surprisingly successful prediction result. This could initiate the potential interests of the public to test out human LLM for tasks not related to human language simply to configure whether human LLM can perform well. This could lead to two potential ethical concerns. (1) Training on LLM is computationally expensive, and rising interest among people trying random training ("giving it a shot" type of approach) on LLM can waste energy and impose environmental burdens. To mitigate this potential issue, it's critical to provide a clear guideline to efficiently structure the training processes for the trial, such as trying out with a small dataset or utilizing the smaller LLM model. Practically, it would be helpful to provide a status bar for energy usage in the cloud system to enhance user awareness of energy usage. (2) The other issue is misinterpretation or overinterpretation of the model which could spread false information. Even though the two approaches resulted in relatively successful predictions compared to the baseline, the attention-based analysis cannot be directly translated into a truly meaningful biological interpretation and requires cautions of claiming associative or causal relationships. If the preliminary analysis is published without experimental or literature validation, it could lead scientists in the field in the wrong direction and share misinformation. To mitigate this issue, it may require in-depth analysis based on other literature or customized experimental validation before making a new biologically meaningful conclusion.

References

[1] Luke et al. Isbel. Generating specificity in genome regulation through transcription factor sensitivity to chromatin. In *Nature Reviews Genetics*, 2022.

[2] Jeewon Yang Dohoon Lee and Sun Kim. Learning the histone codes with large genomic windows and three-dimensional chromatin interactions using transformer. In *Nature Communications*, 2022.

[3] Žiga et al Avsec. Effective gene expression prediction from sequence by integrating long-range interactions. In *Nature Methods*, 2021.

[4] Jiajia et al. Liu. Large language models in bioinformatics: applications and perspectives. In *arXiv*, 2024.

[5] Zhihan et al. Zhou. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. In *arXiv*, 2023.

[6] Chao et al. Yan. Leveraging generative ai to prioritize drug repurposing candidates for alzheimer's disease with real-world clinical validation. In *npj Digital Medicine*, 2024.

[7] Ritambhara Singh et al. Deepchrome: deep-learning for predicting gene expression from histone modifications. In *Bioinformatics*, 2016.

[8] Ritambhara Singh et al. Attend and predict: understanding gene regulation by selective attention on chromatin. In *Adv. Neural Inf. Process Syst.*, 2017.

[9] Chatgpt. https://chatgpt.com/.

[10] Pytorch-transformer. https://pytorch.org/tutorials/beginner/transformer-tutorial.html.

[11] Sp24 hugging face transformers tutorial.

[12] Remap. https://remap.univ-amu.fr/.

[13] Zenab F. Mchaourab et al. Chip-seq and chip-exo profiling of pol ii, h2a.z, and h3k4me3 in human k562 cells. In *Scientific Data by Nature*, 2018.

A Appendix

Learning rate	Batch Size	Early stop with validation	Weighted Loss	Oversampling	MSE
0.0001	32				0.8745
0.0001	64				0.7632
0.0001	128				0.7076
0.00001	128				0.6955
0.00001	128	Yes			0.5712
0.00001	128	Yes	Yes		0.5808
0.00001	128	Yes		Yes	0.5131

Table 3: Hyperparameter tuning and experiment results for Approach I. Part I

model_dim	num_heads	num_layers	ff_dim	Validation Loss
256	2	6	128	0.5052
256	4	6	128	0.5169
64	4	6	256	0.5246
256	8	3	256	0.5246
64	8	3	64	0.5264
64	8	3	128	0.5331
128	2	3	128	0.5345
256	8	3	64	0.5352
64	4	3	128	0.5381
128	8	6	128	0.5382
256	4	6	256	0.5398
256	2	6	256	0.5405
128	4	6	128	0.5453
64	2	6	128	0.5456
256	8	6	256	0.5508
128	2	3	64	0.5532
128	4	3	128	0.5564
128	8	3	64	0.5645
256	4	3	128	0.5681
64	2	3	128	0.5684
64	8	6	128	0.5685
256	2	3	128	0.5695
256	4	6	64	0.5696
256	2	6	64	0.5696
128	2	3	256	0.5708
256	4	3	64	0.5710
64	4	3	64	0.5753
64	8	6	256	0.5760
64	2	6	64	0.5814
64	4	6	128	0.5831
128	8	3	128	0.5908
128	2	6	256	0.5914
128	2	6	128	0.5915
256	2	3	256	0.5981
64	8	6	64	0.6013
128	8	6	256	0.6020
128	4	3	256	0.6063
128	8	6	64	0.6068
64	4	6	64	0.6079
64	2	3	64	0.6079
64	4	3	256	0.6098
128	4	6	256	0.6123
256	8	3	128	0.6135
256	2	3	64	0.6160
256	8	6	128	0.6174
256	8	6	64	0.6190
128	4	3	64	0.6195
64	8	3	256	0.6218
128	2	6	64	0.6248
128	8	3	256	0.6250
64	2	6	256	0.6251
128	4	6	64	0.6323
64	2	3	256	0.6424
256	4	3	256	0.6481

Table 4: Hyperparameter tuning results for Approach I. Part II. (Lr: 0.0001, Batch size: 32. Loss: MSE, With oversampling (minimum 100) and early stopping with patience 10 epoches)

Learning rate	Batch Size	Loss Function	Oversampling	Optimizer	Lr scheduler	Pearson R
0.0001	32	MSE				
0.0001	32	MSE		Adam		0.7647
0.0001	64	MSE		Adam		Out of memory
0.00001	32	MSE		Adam		0.7795
0.0001	32	NRMSE		Adam		0.7649
0.00001	32	NRMSE		Adam		0.7637
0.0001	32	MSE	Yes	Adam		0.8599
0.00001	32	MSE	Yes	Adam		0.8462
0.0001	32	MSE		AdamW	Yes	0.7544
0.0001	32	MSE	Yes	AdamW	Yes	0.8319
0.00005	32	MSE		AdamW	Yes	0.7525
0.00005	32	MSE	Yes	AdamW	Yes	0.8013
0.00001	32	MSE	Yes	AdamW	Yes	0.7520

Table 5: Experimental result for Approach II. With total three epochs