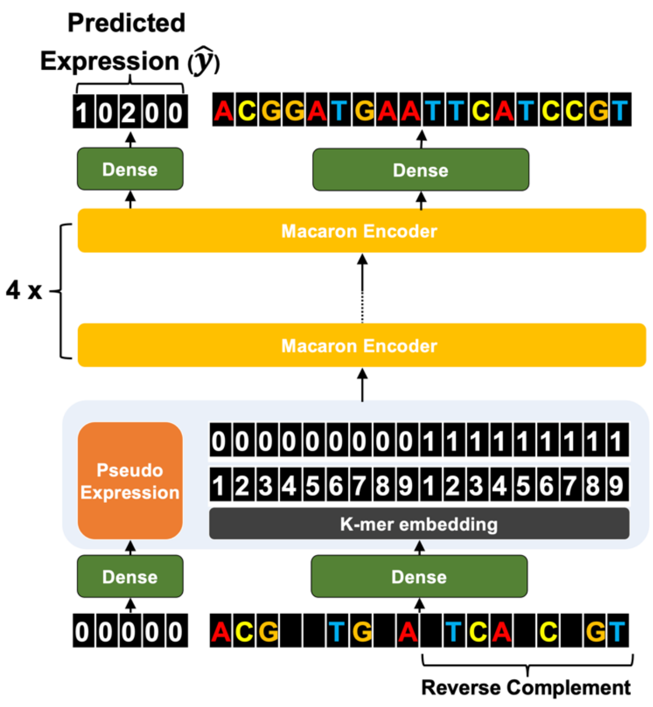
DREAM Challenge 2022
Predicting gene expression using millions of random promoter sequences
by
Team Unlock_DNA

Abstract

The breakthrough high-throughput measurement of the cis-regulatory activity of millions of randomly generated promoters provides an unprecedented opportunity for us to systematically decode the cis-regulatory logic that determines the expression values. In this DREAM challenge, we developed an end-to-end Transformer encoder architecture to predict the expression values from a millions of DNA sequences. Our method introduces several innovative design such as Macaron-like encoder structures, *k*-mer embedding, and pseudo expression values to learn the relationships between a large amount of sequences and expression values. We believe that our model provides a novel method of learning and characterizing how cis-regulatory sequences determine the expression values.

1. Description of data usage

1.1 Sequence trimming and padding
We removed the leading 17 and trailing 13 nucleotides (nt) that are identical in both training and testing promoter sequences, since these nucleotides were not informative for predicting expression values and removing the nucleotides would significantly reduce the training and inference time. The length of the resulting promoter sequences ranged from 48 to 112 nt for training data, while >99.97% training promoters were less than 100 nucleotides. To further reduce the computational overhead, we used 6,737,568 promoter sequences shorter than 100 nt (after trimming) in the model training. For promoters that are less than 100 nt, the left and right sides were padded with letter N.



1.2 Reverse complemented sequences
We empirically found that including the reverse complemented promoter sequences would significantly improve the performance. Thus, the reverse complemented sequences were concatenated with the original sequences (after trimming and padding) and used as the input for model training. Thus, the total length of the input sequences was 200 nt.

1.3 Standardization of the expression values
The expression values were standardized to the mean of zero and standard deviation of one. Our experiments found that when the mean squared error loss was used, standardizing the expression values gave better model generalization performance (in terms of Pearson's R and Spearman's Rho) and faster convergence.

Figure 1. Our model for predicting gene expression from promoter sequences

2. Description of the model

2.1 Macaron-like Transformer encoder

We used a Macaron-like Transformer encoder architecture[1–3] to predict the expression values from promoter sequences (Figure 1). Compared with the regular Transformer encoder, the Macaron-like encoder has two half-step feed forward (FFN) layers at the beginning and the end of each Transformer encoder block (Figure 2 and 3), which can be mathematically interpreted as a numerical Ordinary Differential Equation (ODE) solver for a convection-diffusion equation in a multi-particle dynamic systems. Given the stochastic nature of the input sequences, we hypothesized that this design may better recover the associations between nucleotide pattern and the expression values.

2.2 Convolution layer in the Transformer encoder block

We added a separable 1D convolution layer in the Macaron encoder block following the first FFN layer and in front of the multi-head attention layer. This design has been used in other Transformer architectures such as Conformer[4], and is shown to be critical for capturing the local signals.

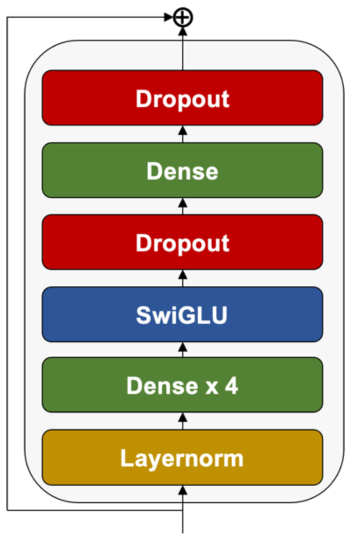


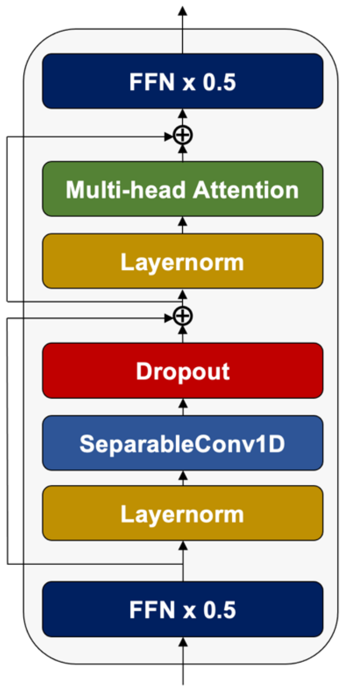
Figure 2. The feed forward (FFN) layer with SwiGLU activation

2.3 K-mer embedding layer

We extracted the sliding *k*-mers from one-hot encoded sequences, and mapped them onto a continuous embedding. It has been previously shown that the *k*-mer embedding of nucleotide sequences had better performance than the convolution on tasks such as predicting transcription factor binding sites[5]. The *k*-mer embedding was then combined with the learnt positional embedding and strand embedding (forward strand vs reverse complemented strand) as one part of the input to the Macaron encoder.

2.4 Pseudo expression positions

Along with the sequence input, we added several positions (32 in our final model) of "pseudo" expression values as the input, where all input expression values were zero's (Figure 1). These "pseudo" expression values were embedded, concatenated with the sequence embedding and fed into the Macaron encoders. Our model predicted one expression value for each "pseudo" expression position, and used the mean of the prediction of all positions as the final predicted expression value. We empirically found that this design had significantly better performance than the conventional design such as using the global pooling layer as the output layer for the regression task.



2.5. Training losses

The training losses consist of the mean squared error between the expression values ($\boldsymbol{y}_{mse}$), and the reconstruction loss ($\boldsymbol{y}_{recon}$), where we randomly masked 5% of the nucleotides and asked the model predict the masked nucleotides. We found that including the reconstruction loss helped stabilize the training process especially with larges models.

Figure 3. The Macaron encoder.

3. Training procedure

Our final model has four layers of Macaron encoder block, with an embedding size of 512, dropout rate of 0.01, *k*-mer size of 10, number of heads in multi-head attention of 8 and 32 "pseudo" expression positions. The final model has around 47 million trainable parameters.

We trained our model on one machine with 4 A100 GPUs. We trained the model on 95% of the training data and evaluated on the remaining 5% of the data. The model with the highest Pearson's R on the testing data was used for the prediction. We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. We varied the learning rate over the source of training according to the formula used in the original Transformer paper[1]. We used a warm up step of 12,500 and a batch size of 512.

4. Contributions and Acknowledgement

4.1 Contributions

Name	Affiliation	Email
Wuming Gong	University of Minnesota	gongx030@umn.edu
Byeong-Chan Kim	Chung-Ang University	
Juhyun Lee	Chung-Ang University	juhyun62015@gmail.com
Il-Youp Kwak	Chung-Ang University	ilyoup.kwak@gmail.com

4.2 Acknowledgement

This work was supported by a grant from the Department of Defense (W81XWH2110606) and Minnesota Regenerative Medicine. We acknowledge the Minnesota Supercomputing Institute for providing computational resources.

5. References

1. Vaswani, A. *et al.* Attention Is All You Need. *arXiv.org* cs.CL, (2017).

2. Press, O., Smith, N. A. & Levy, O. Improving Transformer Models by Reordering their Sublayers. *Arxiv* (2019).

3. Lu, Y. *et al.* Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View. *Arxiv* (2019).

4. Gulati, A. *et al.* Conformer: Convolution-augmented Transformer for Speech Recognition. *Arxiv* (2020).

5. Shen, Z., Bao, W. & Huang, D.-S. Recurrent Neural Network for Predicting Transcription Factor Binding Sites. *Scientific reports* 8, 15270 (2018).