Functional yeast promoter sequence design using temporal convolutional generative language models

Ibrahim Alsaggaf[1] and Cen Wan[1,*]

[1]School of Computing and Mathematical Sciences, Birkbeck, University of London
[*]Correspondence: cen.wan@bbk.ac.uk

Abstract

Promoter sequence design is the key to accurately control gene expression processes that play a crucial role in biological systems. Thanks to the recent community effort, we are now able to elucidate the associations between yeast promoter sequences and their corresponding expression levels using advanced deep learning methods. This milestone boosts the further development of many downstream biological sequence research tasks like synthetic DNA design. In this work, we propose a novel synthetic promoter sequence design method, namely Gen-DNA-TCN, which exploits a pre-trained sequence-to-expression predictive model to facilitate its temporal convolutional neural networks-based generative model training. A large-scale evaluation suggests that Gen-DNA-TCN successfully generated diverse synthetic promoter sequences that also encode similar distributions of transcription factor binding sites to real promoter sequences.

Introduction

Cis-regulatory mechanisms play a fundamental role in biological systems by allowing transcription factors to interpret regulatory DNA sequences and controlling gene expression levels. A long-standing challenge in biology is to accurately control gene expression levels by designing corresponding synthetic DNA sequences. Thanks to the recent community-wide effort [1], we are now able to elucidate the associations between yeast promoter sequences and the corresponding expression levels with the help of advanced deep learning methods. This milestone allows us to further explore the possibility of designing functional synthetic promoter sequences.

The conventional synthetic promoter sequence generation methods are based on pre-defined libraries of known promoters. Redden and Alper (2015) [2] assembled synthetic minimal core promoters using elements that were derived from different libraries. Rajkumar et al. (2016) [3] designed synthetic yeast promoters that are responsive to environmental stress according to known transcription factor binding-associated motifs information. Portela et al. (2017) [4] also designed synthetic promoters by analysing known promoters in libraries. All those conventional methods require task-specific library selection and complex example sequence analysis, on top of the limitations on synthetic sequence generation efficiency. More recently, with the help of deep learning methods, the efficiency and capacity of synthetic promoter design methods have been improved. For example, Kotopka and Smolke (2020) [5] trained an ensemble of convolutional neural networks by using 1 million yeast promoter sequences to design synthetic promoter sequences. However, due to the bottleneck of the sequence-to-expression predictive accuracy (i.e. an average R^2 value of 0.81), the performance and capacity of the model still needed to be improved. Zrimec et al. (2022) [6] used generative adversarial networks to generate synthetic yeast promoter sequences. However, analogously, the sequence-to-expression predictive performance (i.e. R^2 = 0.80) of the associated deep learning model still needs to be further improved. In addition, due to the natural difficulty on GAN training, the computational cost for obtaining the well-trained generative model is extremely high (i.e. 200,000 iterations). In this work,

we exploits a pre-trained sequence-to-expression predictive language model to further train a generative language model. The pre-trained predictive language model obtained a R^2 value of 0.88 over 71,103 high-quality testing sequences, and the well-trained generative model merely takes 32 epochs.

Temporal Convolutional Networks (TCN) [7] is a variant of convolutional neural networks for coping with sequence data. It demonstrated better performance over other popular sequence models like the well-known LSTM [8] and Transformer [9], on a variety of sequence data modeling tasks [10]. In general, TCN is constructed as a type of stacked neural networks, where each hidden layer has the same length as the input layer in order to guarantee that the prediction for the target time point τ_i only depends on the information of all previous time-points (i.e. τ_0, τ_1, τ_2, ..., τ_{i-1}). This type of training paradigm allows TCN to model the probabilistic distributions of all time-points of full sequences via the well-known autoregressive learning approach. TCN has already been used for different sequence data generative tasks [10] like speech generation, text-to-speech generation and music generation. In terms of handling biological sequence data, Shin, et al. (2021) [11] successfully adopted a TCN-based autoregressive model to estimate protein amino acid distributions and generate synthetic protein sequences.

In this work, we present a new TCN-based generative language model, namely Gen-DNA-TCN, which inherits the knowledge of promoter sequence-to-expression associations encoded by P-DNA-TCN – a predictive model that was developed for the recent Random Promoter DREAM Challenge [1]. The P-DNA-TCN model consists of 4.1 million parameters and well approximates the expression levels of random promoter sequences. By using the pre-trained P-DNA-TCN model, we modified its architecture and further trained the Gen-DNA-TCN model with low computational cost. A large-scale evaluation shows that Gen-DNA-TCN successfully generated diverse synthetic promoter sequences that also encode similar transcription factor binding site distributions compared with real yeast promoter sequences. The paper is organized as follows. The Methods section introduces the Gen-DNA-TCN model, followed by the Results section which reports and discusses the performance of Gen-DNA-TCN. The last section presents conclusions and future research directions.

Methods

Overview of Gen-DNA-TCN

In general, the Gen-DNA-TCN model is built on top of a pre-trained predictive temporal convolutional neural networks, namely P-DNA-TCN [12], which consists of 4.1 million model parameters and was trained using 6.7 million random yeast promoter sequences and their corresponding expression levels. According to the recent Random Promoter DREAM Challenge [1] – a large-scale evaluation of deep learning-based sequence-to-expression predictive models, P-DNA-TCN demonstrated good performance in predicting the expression levels of 71,103 testing yeast promoter sequences. The results confirm that P-DNA-TCN successfully captured the associations between regulatory units of yeast promoter DNA sequences and their corresponding expression levels. In terms of Gen-DNA-TCN, we adopted new model training methods but still exploited those 4.1 million parameters of the pre-trained P-DNA-TCN as prior knowledge about regulatory sequence units. Unlike P-DNA-TCN which predicts expression levels, Gen-DNA-TCN sequentially predicts the probabilities of four different nucleotides according to the combinations of any pre-defined nucleotides. By using Gen-DNA-TCN, we successfully generated a pool of 261,664 unique synthetic yeast promoter sequences.

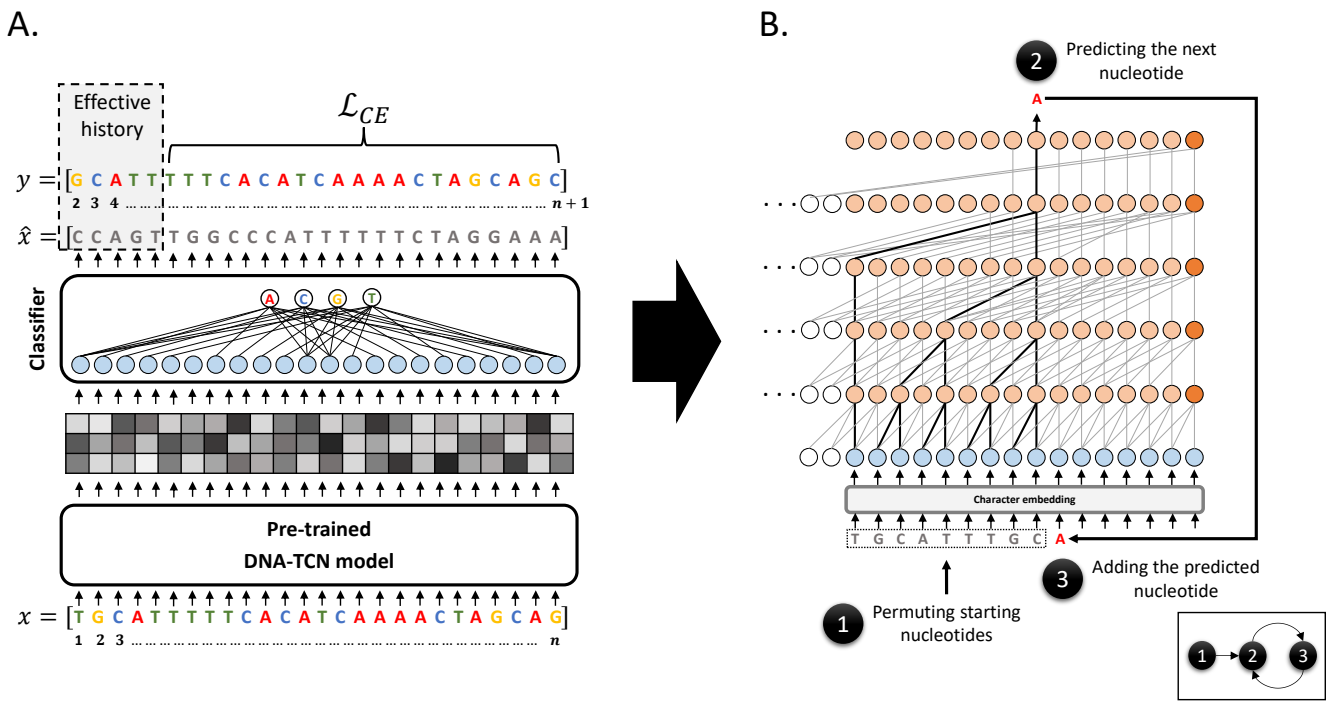


Figure 1: The architecture of Gen-DNA-TCN.

The pre-trained predictive DNA-TCN model

In general, the pre-trained predictive DNA-TCN (P-DNA-TCN) model [12] exploits the conventional character-level TCN [10] architecture to encode the dependencies between sequential nucleotides of DNA sequences. But P-DNA-TCN uses a different output layer, i.e. a linear output layer that is stacked on the last time-point of the top-most residual block. The input layer of P-DNA-TCN first encodes DNA sequences using learnable embedding parameters, by which nucleotides are converted into vector representations. On top of the embedding layer, 5 residual blocks are stacked with incremental dilated convolution sizes (i.e. 1, 2, 4, 8 and 16) to cover entire sequences. In addition to the modified last output layer, P-DNA-TCN also exploits a newly proposed loss function $\mathcal{L}$, as shown in Equation 1. $\mathcal{L}$ includes three terms, i.e. weighted mean squared error (MSE), squared Pearson correlation coefficient, and Spearman correlation coefficient. We used 0.1 as the weight for the term MSE ($w = 0.1$) to reflect the importance of the other two terms, since the expression levels of training and testing DNA sequences follow different scales.

$$\mathcal{L} = w * MSE - (Pearson\, r^2 + Spearman\, \rho) \tag{1}$$

To train the P-DNA-TCN model, we used 6,739,258 random yeast promoter sequences provided in the Random Promoter DREAM Challenge [1]. Those training sequences were put into different groups according to their length. The groups including less than 1,000 sequences were removed from the training dataset, leading to 6,728,720 remaining sequences in 23 groups. During the model training stage, we used the well-known SGD optimizer with a learning rate of 0.05 to conduct the model training. The gradient clipping was also used to facilitate convergence with the maximum norm that was set to 0.15.

The generative DNA-TCN model

To train the generative DNA-TCN (Gen-DNA-TCN) model, we adopted a transfer learning setting – we used the 4.1 million parameters of the pre-trained P-DNA-TCN model as the starting point to train the Gen-DNA-TCN model adopting a different architecture. In terms of Gen-DNA-TCN, the linear output layer of P-DNA-TCN was discarded and a new multi-class classification head was used to predict the next nucleotide according to a set of given nucleotides, as shown in Figure 1A. Let $x = \{x_1, x_2, ..., x_n\}$ be a DNA sequence of length n, where x_i is the ith nucleotide in the sequence. We fed x into the pre-trained P-DNA-TCN model to obtain d-dimensional feature representations for each nucleotide in x, where d equals 384 in this work. Those feature representations are further fed into the multi-class classification head, which consists of a d-dimensional linear layer and an output layer with four nodes representing nucleotides A, C, G, and T, respectively. The output logits are normalised using a Softmax function transforming them into probabilities denoted by $\hat{x}$. We modelled the target sequence y to be the input sequence x but shifted by one nucleotide position, i.e., $y = \{x_2, x_3, ..., x_{n+1}\}$. This changes the task to an auto-regressive prediction task, where each nucleotide x_i in the sequence is predicted given all previous nucleotides $\{x_1, x_2, ..., x_{i-1}\}$. We split $\hat{x}$ and y into effective history (i.e. the first five initial nucleotides) and trained the model on the remaining nucleotides. The well-known cross-entropy $\mathcal{L}_{CE}$ was used as the loss function. As shown in Equation 2, $\hat{x}_{i,y_i}$ is the probability of the target class y_i. The model was trained by 100 epochs using the Adam optimiser with a learning rate of 0.002.

$$\mathcal{L}_{CE} = \frac{1}{n} \sum_{i=1}^{n} -\log \hat{x}_{i,y_i} \tag{2}$$

We used 71,103 random promoter DNA sequences (each with a length of 110 nucleotides) that were used as the highly-quality testing dataset in the Random Promoter DREAM Challenge [1] to train the Gen-DNA-TCN model. Those sequences were further split into a training dataset and a validation dataset with a proportion of 8:2. We then pre-processed those 56,879 training sequences and 14,224 validation sequences by removing the left adapter nucleotides (i.e. TGCATTTTTTTCACATC). After every training epoch, we freeze Gen-DNA-TCN and measure the cross entropy value on the validation sequences. After completing the 100 training epochs, we selected the model that obtained the lowest cross entropy value on the validation sequences.

Table 1: The numbers of promoter sequences in different sequence sets.

Bin Index	Expression Range	All Real	Training	Validation	All Synthetic	Sampled Synthetic
1	3.03 - 4.52	22	17	5	0	0
2	4.52 - 6.02	652	521	131	12	12
3	6.02 - 7.52	4,420	3,536	884	597	597
4	7.52 - 9.02	9,298	7,438	1,860	10,377	7,438
5	9.02 - 10.52	14,910	11,928	2,982	46,991	11,928
6	10.52 - 12.02	13,754	11,003	2,751	74,421	11,003
7	12.02 - 13.52	11,659	9,327	2,332	67,955	9,327
8	13.52 - 15.02	10,352	8,281	2,071	45,271	8,281
9	15.02 - 16.52	5,611	4,488	1,123	14,471	4,488
10	16.52 - 18.02	425	340	85	1,513	340
11	18.02 - 19.56	0	0	0	56	0
Total	3.03 - 19.56	71,103	56,879	14,224	261,664	53,414

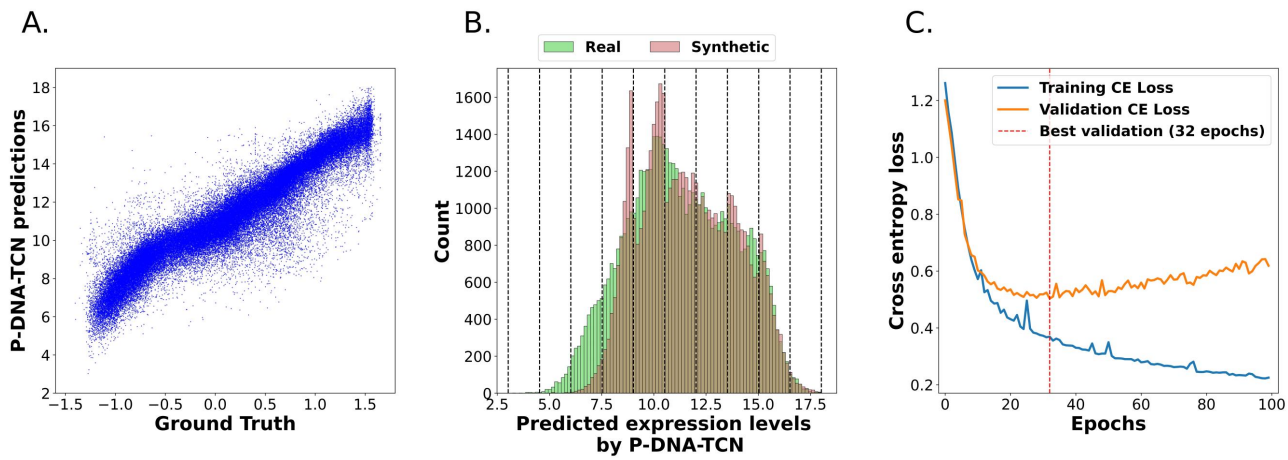


Figure 2: A. The pairwise comparisons of true and P-DNA-TCN-predicted expression levels of 71,103 real promoter sequences. B. The histograms of P-DNA-TCN-predicted expression levels of 56,879 real promoter sequences and 53,414 sampled synthetic promoter sequences. C. The cross entropy loss function curves of the training and validation sequence sets over 100 epochs.

Synthetic promoter sequence generation strategies

To generate synthetic promoter sequences, we first created 262,144 starting sequences of length 9 nucleotides using all possible permutations of the four nucleotides (i.e. A, C, G, and T). Those starting sequences were fed into the trained Gen-DNA-TCN model to predict the next nucleotide. After predicting the 10th nucleotide, we append it to the starting sequence and pass the new sequence (i.e. starting sequence + predicted 10th nucleotide) again into the model to predict the 11th nucleotide. This process was iterated until the generated sequence length equals the pre-processed training sequence length excluding the starting sequence. We then discarded the starting sequences and added the left adapter at the beginning of the generated sequences making their length equal to 110. Finally, we filtered out any duplications and any matches in the 71,103 training sequences of Gen-DNA-TCN and the 6.7 million training sequences of the pre-trained P-DNA-TCN model, leading to 261,664 unique generated synthetic promoter sequences.

Results

Gen-DNA-TCN successfully learned nucleotide dependencies in the 56,879 real yeast promoter sequences

We first evaluated the expression levels of the generated synthetic promoter sequences by using the P-DNA-TCN model. As mentioned in the Introduction section, according to the recent Random Promoter DREAM Challenge [1] results, P-DNA-TCN obtained a squared Pearson correlation of 0.88 and a Spearman correlation of 0.94 on 71,103 high-quality real yeast promoter sequences. Figure 2.A shows the pairwise comparisons between the true and P-DNA-TCN-predicted expression levels, suggesting that P-DNA-TCN has successfully learned the sequence-to-expression associations of yeast promoters.

By using the weights of the pre-trained P-DNA-TCN model, we successfully trained the Gen-DNA-TCN model and generated 261,664 unique synthetic promoter sequences. According to the range of the P-DNA-TCN-predicted expression levels for both real and synthetic sequences (i.e. 3.03 - 19.56), we created 11 bins and randomly sampled 53,414 synthetic promoter sequences from different bins except the first and the last bins, since all the synthetic sequences

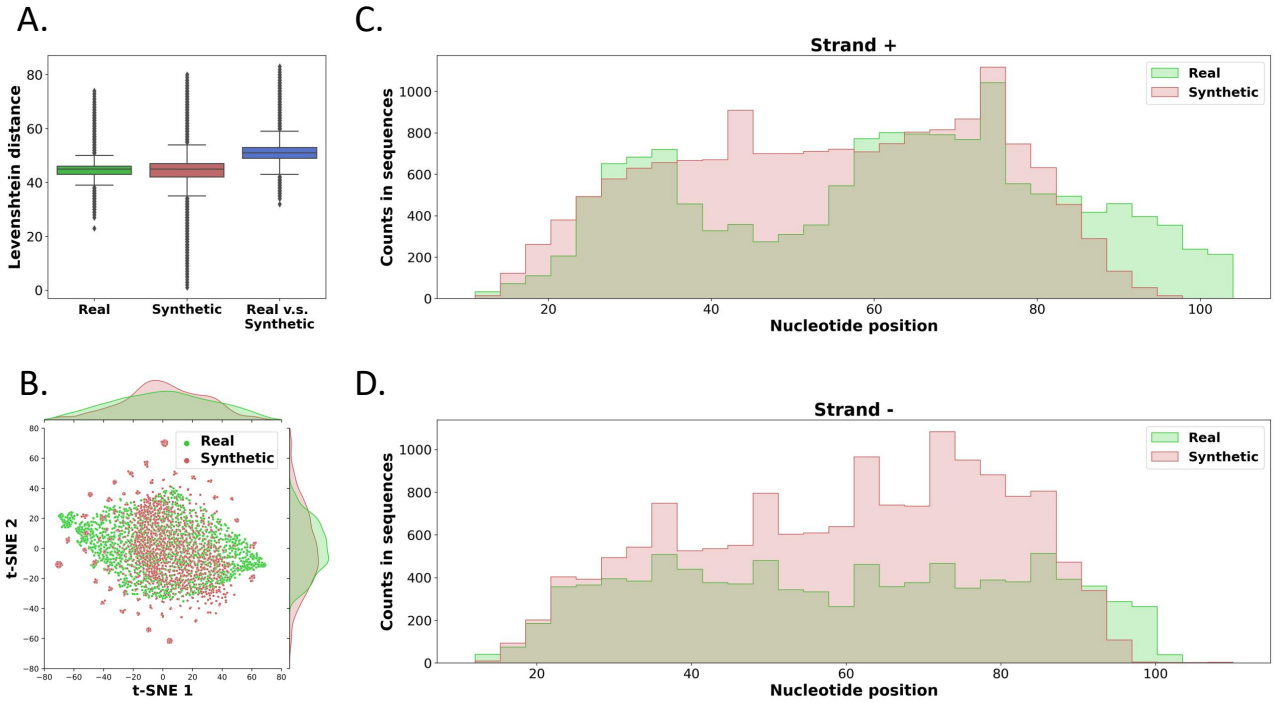


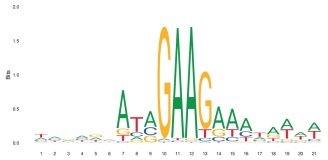
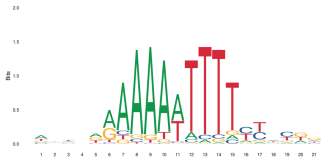
Figure 3: A. The boxplots showing the distributions of pairwise Levenshtein distance of sequences in different sets. B. The 2D t-SNE visualisation of the real and sampled synthetic promoter sequences. C. - D. The distributions of the significant positive and negative strands-based transcription factor binding sites encoded in the real and sampled synthetic promoter sequences.

whose predicted expression levels are greater than 4.52 and there is no real sequence whose predicted expression level is greater than 18.02. Table 1 shows the details about the number of sequences in different groups. Figure 2.B shows two histograms denoting the P-DNA-TCN-predicted expression levels of the 56,879 real (training) promoter sequences and those 53,414 sampled synthetic promoter sequences. In general, the green and red histograms have a large overlapping area, suggesting that the majority of the sampled synthetic promoter sequences lead to similar expression levels to the real promoter sequences. In addition, the expression profiles of both real and synthetic promoter sequences follow a similar range. The range of the former's expression levels is between 3.23 and 18.02, whilst the latter's expression levels range between 4.94 and 17.97. It suggests that the trained Gen-DNA-TCN model can create a pool of synthetic promoter sequences that covers a similar range of expression levels of the real promoter sequences. Figure 2.C shows the curves of the loss function obtained by the training and validation promoter sequences. The optimal Gen-DNA-TCN was obtained after merely 32 epochs of training, suggesting a good computational efficiency.

Gen-DNA-TCN successfully generated diverse synthetic promoter sequences and the distributions of the yeast transcription factor binding sites encoded in the synthetic promoter sequences are similar to the ones in the real promoter sequences

We further evaluated the differences between the real and synthetic sequences and compared their transcription factor bind site distributions. Figure 3.A shows three boxplots about the distributions of pairwise Levenshtein distance [13] obtained by three groups of sequences. The green boxplot shows the pairwise Levenshtein distance between all 56,879 real (training) promoter sequences, whilst the red boxplot shows the pairwise Levenshtein distance between those 53,414 sampled synthetic promoter sequences. Although both boxplots show the same median Leven-

Table 2: Example synthetic promoter sequences generated by Gen-DNA-TCN.

Predicted Expression Level	Generated Synthetic Sequence	Transcription Factor Binding Site	P-value	Sequence Logo
5.84	TGCATTTTTTTCACATCGGTGATCTAGCCTTAAA GTATGGAGTAATTAAGGGTTGGCGGCTAATTTA AGGGTCTTATTAGGGGGGGGTGGTTACGGCTG TTACGGTTACG	UME6	1.27e-07	
12.4	TGCATTTTTTTCACATCGCACTTAACAAAAAGTA AGAACTCCTATATAGAAGATATAAAAGAAAGA CAATACTCACTTATATACCAAGGGTTACGGCTG TTTTGTTTAAG	SFL1	1.48e-07	
17.1	TGCATTTTTTTCACATCCGGGTAATACTTTCAGG CTCAAGCCCTGAGTATGCTGGGTGGTAATGCC AGCGCTTAAAAAATTTTTTTTCGGTTACGGCTG TTTAGGTTGAT	SFP1	1.91e-08	

shtein distance (i.e. 45), the interquartile range of the synthetic sequences is wider than the one for the real sequences, suggesting that those generated synthetic sequences are more different to each other, compared with the case in the real promoter sequences. The blue boxplot shows the distributions of Levenshtein distance obtained by the pairwise comparisons between the real and sampled synthetic sequences. It is obvious that the real and synthetic sequences are different to each other, due to the median Levenshtein distance value of 51. Figure 3.B further illustrated the 2D t-SNE [14] visualisation of both the real and sampled synthetic sequences. It is clear that the green and red dots are distributed in different areas in general, suggesting that Gen-DNA-TCN is able to generate diverse synthetic sequences w.r.t. the real sequences.

We then compared the distributions of the transcription factor binding sites encoded in the real and sampled synthetic promoter sequences. We used the well-known FIMO motif searching tools [15] and the JASPAR database [16] to detect significant transcription factor bind sites. Due to the large promoter sequences sample size, we adjusted the significance levels as 4.40e-07 and 4.68e-07 respectively for the real and sampled synthetic sequences by using the stringent Bonferroni correction. Figures 3.C and 3.D show the distributions of the detected transcription factor binding sites on both the real and synthetic promoter sequences. It is clear that those sampled synthetic promoter sequences encode similar transcription factor binding site information to the real sequences, suggesting that the trained Gen-DNA-TCN model can successfully generate functional promoter sequences. Table 2 shows 3 example synthetic promoter sequences with their significant positive-strand transcription factor binding sites.

Conclusion

In this work, we proposed a novel temporal convolutional networks-based DNA sequence language model to generate diverse yeast synthetic promoter sequences that bear gene expression regulatory functions. Future research would be focused on further investigating the associations between synthetic sequence diversity and the length of generation starting nucleotides for the Gen-DNA-TCN model. In addition, the proposed methods could be extended to model more complex DNA sequences.

References

[1] Abdul Muntakim Rafi, Daria Nogina, Dmitry Penzar, et al. A community effort to optimize sequence-based deep learning models of gene regulation. *Nature Biotechnology*, 2024.

[2] Heidi Redden and Hal S. Alper. The development and characterization of synthetic minimal yeast promoters. *Nature Communications*, 6(7810), 2015.

[3] Arun S. Rajkumar, Guodong Liu, David Bergenholm, et al. Conformational scaling relations of two-dimensional macromolecular graphene oxide in solution. *Nucleic Acids Research*, 44(17):e136, 2016.

[4] Rui M. C. Portela, Thomas Vogl, Claudia Kniely, et al. Synthetic core promoters as universal parts for fine-tuning expression in different yeast species. *ACS Synthetic Biology*, 6(3):471–484, 2017.

[5] Benjamin J. Kotopka and Christina D. Smolke. Model-driven generation of artificial yeast promoters. *Nature Communications*, 11(2113), 2020.

[6] Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, et al. Controlling gene expression with deep generative design of regulatory dna. *Nature Communications*, 13(5099), 2022.

[7] Aaron van den Oord, Sander Dieleman, Heiga Zen, et al. Wavenet: A generative model for raw audio. *arXiv:1609.03499*, 2016.

[8] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *arXiv:1706.03762*, 2017.

[10] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[11] Jung-Eun Shin, Adam J. Riesselman, Aaron W. Kollasch, et al. Protein design and variant prediction using autoregressive generative models. *Nature Communications*, 12(2403), 2021.

[12] Ibrahim Alsaggaf et al. Dream challenge submission report of team. `https://github.com/de-Boer-Lab/random-promoter-dream-challenge-2022/blob/main/dream_submissions/Wan%26Barton_BBK/report.pdf`, 2022.

[13] Max Bachmann. Levenshtein. `https://github.com/rapidfuzz/Levenshtein`, 2024.

[14] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[15] Charles E. Grant, Timothy L. Bailey, and William Stafford Noble. Fimo: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[16] Aziz Khan, Oriol Fornes, Arnaud Stigliani, et al. Jaspar 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Research*, 46(D1):D260–D266, 2018.