USING SELF-ATTENTION CONVOLUTIONAL FEATURES AND AUTO ENCODER TO PREDICT ENHANCER-PROMOTER INTERACTIONS

Ziheng (Leo) Li
zl2990

Daniel Lee
jdl2194

Thang Nguyen
tn2468

ABSTRACT

In biology, transcription is the process of copying DNA into RNA by an enzyme called RNA polymerase in order to regulate gene expression. Specifically, RNA polymerase focuses on transcribing regions of DNA called genes. However, with the human genome having 3.2 billion base pairs long, locating said regions is not trivial. In order to facilitate the process, the enzyme leverages promoters, DNA sequences at the beginning of genes that mark the start of the transcription process. In addition, DNA also contains enhancers sequences, which are located thousands of base pairs away from promoters and contain activator proteins that boost RNA polymerase's efficiency. The interactive property between promoters and enhancers and its tie to gene expression has remained an open question, with many researches focusing on determining the relation between the sequence structures of enhancers, promoters and their interactions. In this research, we propose two machine learning models to identify enhancer-promoter interactions (EPI). We present EPSAT, a deep learning model based on SPEID (Sequence-based Promoter-Enhancer Interaction with Deep learning)[1] with an enhancement of self-attention approach from SATORI[2] , and EPLAE, a novel deep learning logistic variational auto-encoder architecture model. Our results for EPSAT achieve higher F_1 score than SPEID, while having lower count of trainable parameters and epochs. The models can be used for not only to predict EPI in DNA, but also provide a general method for evaluating the effects of sequence modification in gene expression.

1 Introduction

Human genome is encoded in chromatins, which are, together with histones packed into highly sophisticated 3-dimensional structures. The way with which different genes are expressed or inhibited in different conditions (cell types, disease) are yet to be fully understood. Only a fraction ($< 2\%$) of genomes in the chromosome are expressed. Previous belief is that the non-gene regions are reminiscent of evolution. Recent studies show these regions play important roles in regulating what genes get transcribed into mRNA. Methods such as ChIP-PET, ChIP-Seq gave rise to study how different regions of the chromosome interact [3]. For this project, we are focusing on tackling the idea behind Enhancer-Promoter interaction (EPI) in order to assess their ability to regulate gene expressions. In order to pursue this objective, we leverages the computing power of deep learning. Within the space of transcription factors cooperativity and EPI research, deep learning methods have been utilized due to their ability to capture non-linear feature interactions that explain underlying regulatory phenomenon [2], alongside with multiple classification and regression problems such as transcription factor prediction, chromatin accessibility analysis, prediction of chromatin structure and its modification, and identification of RNA-binding sites.

Specifically, the project starts by exploring SPEID (Sequence-based Promoter-Enhancer Interaction with Deep learning)[1], a deep learning model focuses on detecting interaction between enhancers and promoters pairs. The model consists of convolutional layers branches on both DNA sequences, a Long Short Term Memory (LSTM) layer to enforce context, and a final dense layer for binary classification. We enhance this architecture with self-attention layer as inspired by SATORI [2], a self-attention with convolutional recurrent neural network (RCNN) model that captures regulatory element interactions in gene sequences. Therefore, we first propose EPSAT, Enhancer-Promoter

Self-Attention Network that will learn the interaction pattern between the enhancers and the promoters. See Figure 1 for the model architecture. The model shows a better performance than our baseline models, SPEID and TargetFinder [4, 5]. We also propose EPLAE, Enhancer-Promoter auto-encoder network for predicting interactions between the enhancers and the promoters. Auto-encoder has been widely used as neural representation learning and are notable in their interoperability for the original data [6]. We show that the EPLAE model can be used to locate interacting sequences while retaining high classification accuracy. See Figure 2 for the architecture. Both the EPSAT and the EPLAE model outperform SPEID [1], achieving an average f1-score of 0.910, and 0.943, respectively, across cell-lines, while SPEID has an average f1 of 0.833.

2 Methods

2.1 Enhancer-Promoter Sequences

We use the Enhancer-Promoter Interaction dataset used in SPEID paper [5]. The data include six cell lines, namely GM12878, HeLa-S3, HUVEC, IMR90, K562, and NHEK. The enhancers and promoters are specific to the cell-lines. They have been identified using annotations from the ENCODE Project [7] and Roadmap Epigenomics Project [8]. The dataset consists of each cell line's enhancer-promoter pairs with a label of 1 if interacting or a 0 if non-interacting. This was identified using high-resolution genome-wide measurements of chromatin contacts in each cell line based on Hi-C [9]. The input sequence is one-hot encoded. The promoter sequence is 2000 base-pairs(bps) long and the enhancer 3000.

The data set is naturally very imbalanced due to the nature of greater prevalence of the non-interacting enhancer-promoter pairs compared to the interacting ones. When an algorithm receives significantly more examples of negative class, it will become biased towards predicting the negative class and fail to learn the underlying pattern between each class. There are a variety of ways to address an unbalanced data set. Data augmentation was performed in order to increase the number of positive data points. Due to the fact that the labels are invariant to small shifts in the sequences, the provided data was augmented to by randomly shifting positive promotes and enhancers to create more data points to address the data imbalance issue [1]. For this paper, we simply choose a positive example and make 20 more copies. This is because 20 negative pairs were sampled per positive pair [9].

Table 1: Positive sample, augmented positive sample, and negative sample counts, for each cell line.

Cell Line	Positive Pairs	Augmented Positive Pairs	Negative Pairs
GM12878	2,113	42,260	42,200
HeLa-S3	1,740	34,800	34,800
HUVEC	1,524	30,480	30,400
IMR90	1,254	25,080	25,000
K562	1,977	39,540	39,500
NHEK	1,291	25,820	35,600
Total	9,899	197,980	197,500

2.2 Data and Code

Code | Data.

2.3 Hardware

All the models are trained locally on sysmtems with NVIDIA GeForce RTX 3090, 128GB memory, and Intel i9-10900KF processor.

2.4 Model Architecture

2.4.1 EPSAT

EPSAT is SPEID with a self-attention layer added before the last dense layer. Therefore, we first create convolutional neural network for each enhancer and promoter sequences to extract sequence motifs. We use 200 kernels with filter length of 13, padding of 0, and stride of 1. 1-dimensional batch normalization is done for faster convergence [10]. ReLU is used as the activation function and the outcome is max-pooled with size of 6 and dropped at rate of 0.2. This allows for extraction of the most prominent features with robustness in spatial shifts and reduces dimensionality. After motif

learning at convolutional network, we concatenate them to form a combined representation of the enhancer-promoter sequence.

The concatenation is then fed to a bi-directional Long Short-Term Memory (LSTM) layer to learn the extracted subsequence features in the context of the sequence as a whole. Hidden size of 100 is used with 2 layers. It is followed by a dropout layer at 0.4 rate to prevent overfitting.

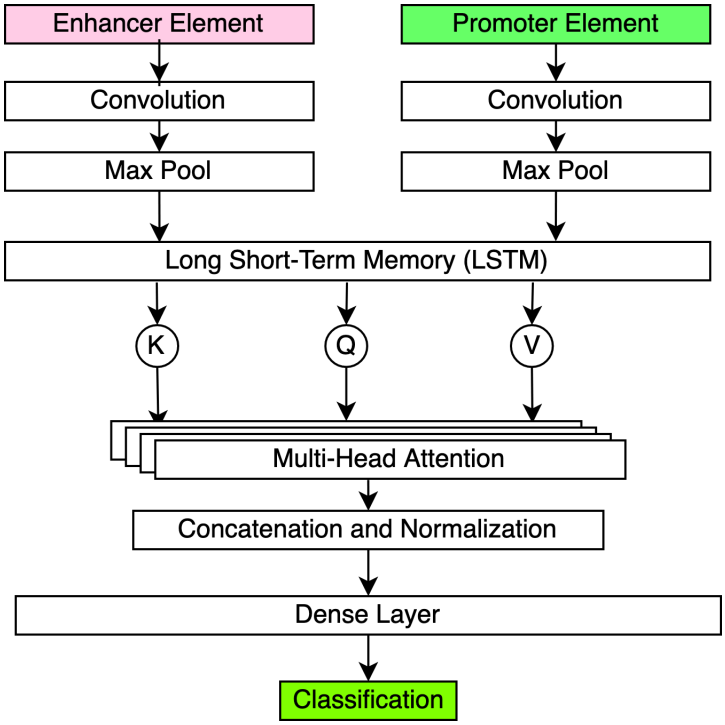

Figure 1: Model architecture: we use a convolutional layer followed by LSTM, a multi-head self-attention layer. The input in both cases is a one-hot encoding of the DNA sequence. The output of the model is a binary prediction

The output is fed to our multi-head self-attention layer. This allows for information capturing between base pairs within the input sequence regardless of the distance [11]. 8 multi-heads are used with a single-head size of 32. The multi-head size of 100 is used for the output of attention layer. As the core part of EPSAT is this multi-head self-attention network, we provide the following detailing.

Given input X, two linear transformations Query Q and Key K are defined as:

$$Q = W_Q^\top X, \tag{1}$$

$$K = W_K^\top X, \tag{2}$$

where W_Q and W_K are the corresponding weight matrices for the Query and Key. Then the attention matrix A is computed as such:

$$A(Q, K) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right), \tag{3}$$

where d_k is the K's dimension. The attention matrix A is a $d \times d$ matrix where $d = \frac{L}{M_p}$, L is the length of the input sequence, and M_p is the size of the window used in max-pooling. For every row in d, each position represents the influence other positions have on that position. The dot product between Q and K shows the relevance of all other positions in the corresponding row of attention matrix. The output matrix V is generated by:

$$V = W_V^T X \tag{4}$$

where W_V is the associated weight matrix. Then the output of self-attention layer is matrix multiplication of A and V.

$$Z = A \times V \tag{5}$$

Through back-propagation step, these weight matrices will learn which inter-dependencies are relevant and what are not. The Q, K, and V heads are concatenated followed by linear transformation. The final output is fed through a multi-head ReLU function, collapsed along the attention layer dimensions through addition and normalized.

The last layer is a dense network units with 2 output states of No-Interaction and Yes-Interaction. A softmax function is used to return the probability for each class. The first value represents the probability of the sequence having no interaction and the second value having an interaction.

During the training, we use the standard cross entropy loss function and Adam optimizer with learning rate of 0.001. On top, a batch size of 64 and an epoch size of 50 are used on the 90 percent of the whole dataset for training.

Note that the hyper-parameters are picked to reduce the number of model parameters to achieve a high representation rate. The EPSAT architecture is partial inspired by SATORI [11].

2.4.2 EPLAE

Lastly, we create our novel model EPLAE to introduce a new approach to sequence representation. The encoder uses four consecutive convolutional layers, each to down-sample the input by half. Following convolution is a vanilla dense layer outputting the latent space. The decode has the same architecture as the encoder except the convolutional layers are transposed, upsampling the latent representation to reconstruct the original sequence. The model architecture is given in Figure 2.

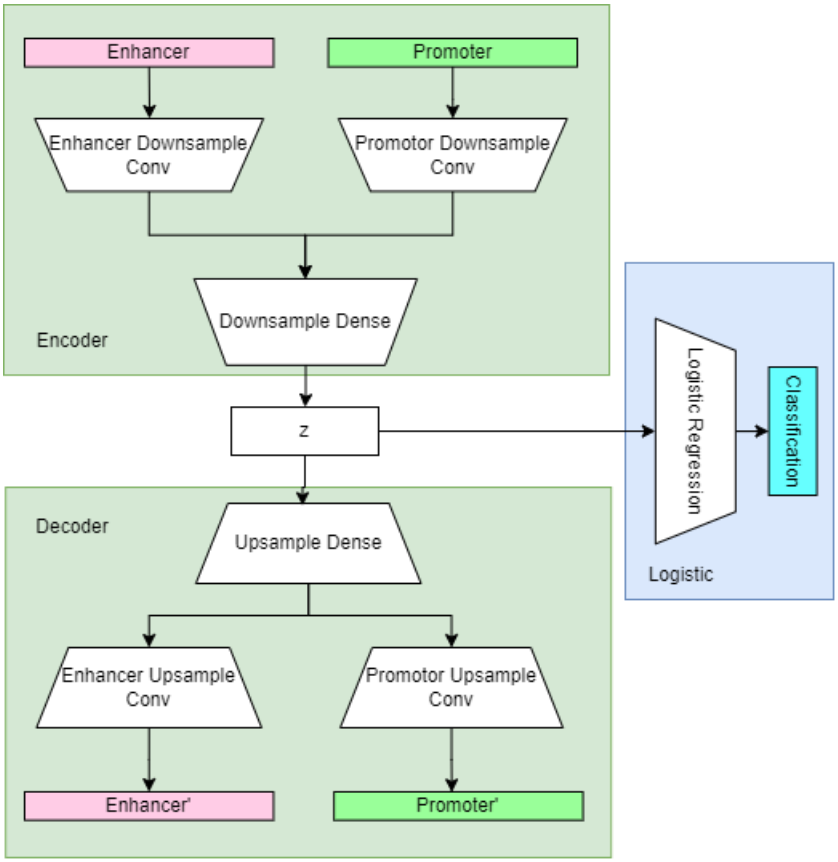

Figure 2: Model architecture: we use a convolutional layer followed by LSTM, a multi-head self-attention layer. The input in both cases is a one-hot encoding of the DNA sequence. The output of the model is a binary prediction

The latent space is used by a logistic regression to classify interacting vs. non-interacting sequences. The logistic model is trained along side the auto-encoder. We added the classification cross-entropy to the reconstruction loss to encourage disentanglement of the interacting state in the latent space.

$$z = F_{encoder}(\theta; x), \hat{x} = F_{decoder}(\phi; z), \hat{y} = F_{logistic}(\psi; z) \tag{6}$$

, where z is the latent representation of input x, $\hat{x}$ is the reconstruction sample. $\hat{y}$ is the predicted label for interacting or non-interacting. θ, ϕ, ψ are the parameters of the encoder, decoder, and logistic model respectively. The loss function encapsulating reconstruction and classification is given as follows.

$$\mathcal{L} = ||x - \hat{x}||^2 + CE(y, \hat{y}) + \lambda_{l2}(||\theta||^2 + ||\phi||^2 + ||\phi||^2), \tag{7}$$

where CE is the cross-entropy loss. y is the true label for interaction. λ_{l2} is the weight of the l2 regularization, set to 1e-5 to avoid overfitting and exploding gradient.

Table 2: Neural networks structure for SPEID and EPSAT.
fc = fully connected layer, conv = convolution layer, Multi-head Attention <head size> * <number of attention heads>

Layers	SPEID	EPSAT	EPLAE
1	conv1 200	conv1 200	Conv1-3 256
2	dropout 0.2	dropout 0.2	Dense 512
3	lstm 200	lstm 200	Conv4-7 256
4	dropout 0.2	dropout 0.2	Logistic-dense 512
5	flatten	flatten	-
6	fc - sigmoid	multi-head attention 32*8	-
7	-	fc - softmax	-
Number of Parameters	19,848,603	685,070	85,619,200+1026

3 Results

3.1 Metric

Our evaluations are done with F1-score, and both EPSAT, EPLAE have outperformed the benchmarks. F1-score is defined to be

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}, \tag{8}$$

where $precision$ is the fraction of true positives among all positives and $recall$ is the fraction of true positives among false negatives and true positives. Essentially, the F1 score allows comprehensive comparison between models that have varying precision and recall performances. The higher the F1-score, the better the model is performing.

3.2 EPSAT

EPSAT outperformed SPEID and TargetFinder in all six cell lines as shown in Table 3. Compared to SPEID whose total parameters are 19,850,805, EPSAT had fewer parameters of 685,070. Nevertheless, EPSAT performed better with 50 epochs compared to SPEID's 360 epochs. The loss history during training and validation is shown in Figure 3.

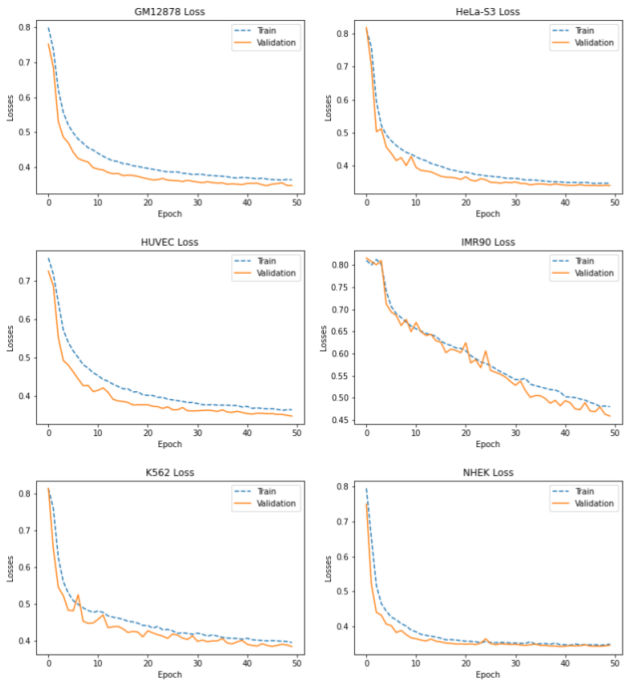

Figure 3: Losses of training and validation for the six cell lines for EPSAT.

3.3 EPLAE

The f1 score of the EPLAE model closely follows that of EPSAT. It outperforms the baseline models SPEID and TargetFinder. The model converges at 50 epoch with l2 regularization set at 1e-5.

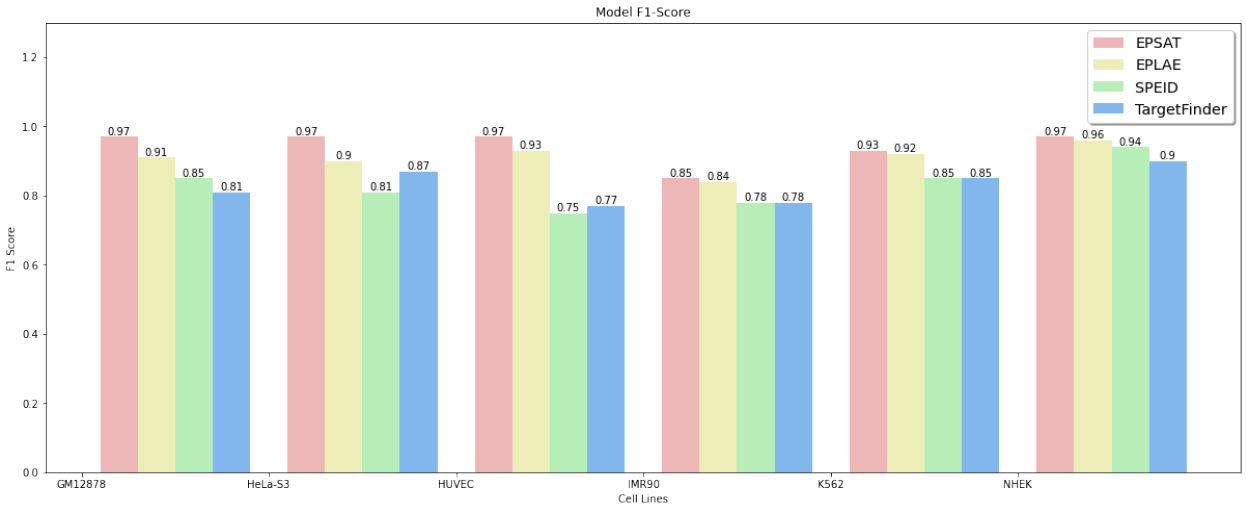

Figure 4: F1-Scores of the six cell lines across EPSAT, EPLAE, SPEID, and TargetFinder E/P/W.

Table 3: F_1 scores of different EPI prediction methods for each cell line.

Model	GM12878	HeLa-S3	HUVEC	IMR90	K562	NHEK
EPSAT	0.97	0.97	0.97	0.85	0.93	0.97
EPLAE	0.91	0.90	0.93	0.84	0.92	0.96
SPEID	0.85	0.81	0.75	0.78	0.85	0.94
TargetFinder (E/P)	0.59	0.61	0.48	0.48	0.61	0.82
TargetFinder (EE/P)	0.84	0.83	0.71	0.83	0.81	0.83
TargetFinder (E/P/W)	0.81	0.87	0.77	0.78	0.85	0.90

4 Discussion

4.1 Model Comparison

In this paper, we investigate the biological phenomenon of regulatory elements interaction that occurs during transcription. We utilize machine learning algorithms to leverage nonlinear modeling to study this interaction. In particular, we examine the interaction between the enhancers and the promoters in six human ENCODE cell lines, namely GM12878, HeLa-S3, HUVEC, IMR90, K562, and NHEK [8]. We first propose EPSAT, Enhancer-Promoter Self-Attention network, inspired by SPEID and SATORI models, and EPLAE, Enhancer-Promoter Logistic Autoencoder network. Enhancer-Promoter data has been collected from TargetFinder paper[5]. The evaluation of the models have been made using F1-Score metric across our models and the benchmark models, namely SPEID and TargetFinder.

SPEID model can be summed up as two convolutional layers each motif-extracting a promoter sequence or an enhancer sequence followed by a bidirectional LSTM layer on the concatenated enhancer-promoter features to predict whether there will be an interaction or not. TargetFinder uses boosted tree approach to predict such an interaction. TargetFinder comes in 3 variants, which utilizes data from different regions. Enhancer/Promoter (E/P) uses only the annotation of the enhancer and Promoter. Extended Enhancer/Promoter (EE/P) uses enhancer annotations with an extended 3kbp flanking region. Lastly, Enhancer/Promoter/Window (E/P/W) utilizes the region/window between the enhancer and promoter. The below tables demonstrate the F_1 scores for different cell types with each respective model.

Our EPSAT has outperformed SPEID and TargetFinder. This has a great value-add as our model is about 29 times smaller than SPEID in parameters yet achieves a greater F1-score in one-eighth of SPEID's epoch size. This shows that the multi-head self-attention layer has very efficiently learned the features with fewer parameters and the context of the sequence was not fully captured by the bi-LSTM layer alone. This suggests that SPEID model can improve itself by adding a multi-head self-attention layer on top of its recurrent layer.

Our EPLAE also yields higher classification accuracy than the baseline models. In Figure 5, we show that the interacting and non-interacting sequences' latent space shows visual different. It implies that the latent space is disentangled by interaction property of the enhancer-promoter pairs. The PCA space also shows separation base on interaction comparing to the original BP sequences.

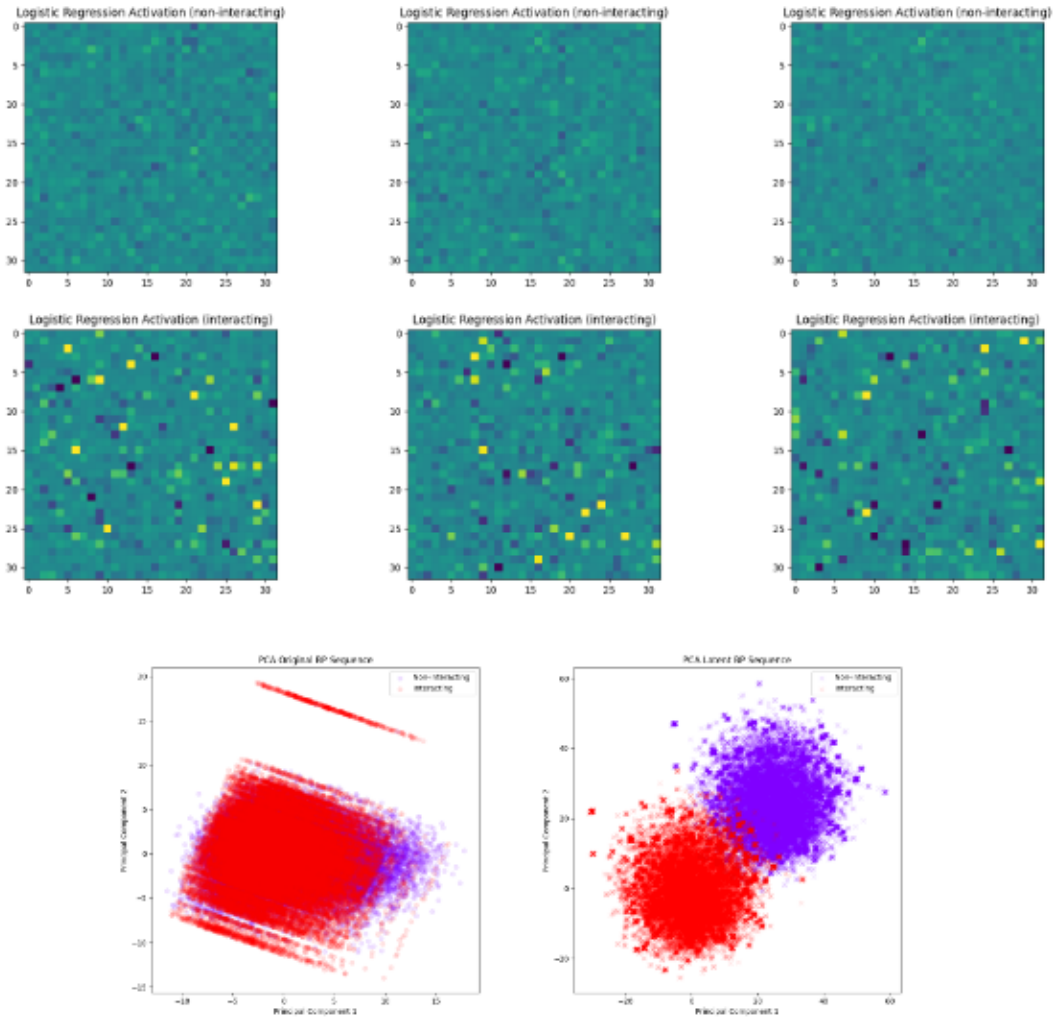

Figure 5: Above: example activation of the logistic regression layer of six samples (three interacting and three on-interacting). Below: PCA of the original data space and the encoded latent space.

4.2 Challenges

The challenge we faced was the unbalanced dataset. As mentioned above, 20 negative samples were collected per 1 positive sample. Therefore, we had to figure out how to address the bias the dataset will present to the model. We re-examined the paper and decided to augment the dataset like SPEID. This was a good reminder to always augment your minor class to allow a model to learn patterns between classes. For EPLAE, we had initially experimented with -VAE in the hope the model can disentangle interaction unsupervised. However, the -VAE model suffers from large reconstruction error when the value is high, while the KLD diverges when the value is reduced. Noting more time than allotted would need to be spent on creating a working -VAE to account for the discreteness of the DNA sequence. We switch to regular auto-encoders added with the logistic function and obtained comparable results as aforementioned.

4.3 Next Step

One potential step to take is to explore what other problem this architecture can solve. Applying this model for other transcription factor interaction problem will be a valuable future direction. Experimenting with different auto-encoder structures is another interesting direction. It is up to debate whether unsupervised approach can resolve the interaction

property in enhancer and promoter. While we find the generative -VAE requires addition work. We want to note a slight modification of the proposed EPLAE can be adding KLD in the loss function. Doing so will encourage a sparse latent representation, which may result in more disentangled latent space than shown in Figure 5. Given the time constraint, we have not able to back-project the latent variables to the original data space. The back projection may provide valuable insight as to the interacting sequences at micro-level.

5 ACKNOWLEDGEMENTS

We want to acknowledge Professor David Knowles for the guidance given to us in class and the TAs for evaluating our project reports. Without them, we would not have explored the project idea to the depth that we were able to.

5.0.1 Conflict of interest statement.

None declared.

References

[1] Barnab225;s P243;czos Jian Ma Shashank Singh, Yang Yang. Predicting enhancer-promoter interaction from genomic sequence with deep neural networks. *Quantitative Biology*, 7(2):122, 2019.

[2] Fahad Ullah and Asa Ben-Hur. A self-attention model for inferring cooperativity between regulatory features. *Nucleic acids research*, 49(13):e77–e77, 2021.

[3] Zhonghui Tang, Oscar Junhong Luo, Xingwang Li, Meizhen Zheng, Jacqueline Jufen Zhu, Przemyslaw Szalaj, Pawel Trzaskoma, Adriana Magalska, Jakub Wlodarczyk, Blazej Ruszczycki, et al. Ctcf-mediated human 3d genome architecture reveals chromatin topology for transcription. *Cell*, 163(7):1611–1627, 2015.

[4] Shashank Singh, Yang Yang, Barnabás Póczos, and Jian Ma. Predicting enhancer-promoter interaction from genomic sequence with deep neural networks. *Quantitative Biology*, 7(2):122–137, 2019.

[5] Sean Whalen, Rebecca M Truty, and Katherine S Pollard. Enhancer–promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature genetics*, 48(5):488–496, 2016.

[6] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[7] EA Feingold and L Pachter. The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640, 2004.

[8] Bradley E Bernstein, John A Stamatoyannopoulos, Joseph F Costello, Bing Ren, Aleksandar Milosavljevic, Alexander Meissner, Manolis Kellis, Marco A Marra, Arthur L Beaudet, Joseph R Ecker, et al. The nih roadmap epigenomics mapping consortium. *Nature biotechnology*, 28(10):1045–1048, 2010.

[9] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.