**DREAM Challenge 2022**

**Predicting gene expression using millions of random promoter sequences**

**by**

**DrAshokAndFriends**

***Abstract***

Translation of DNA to protein via RNA is the key information decoding pathway in the logic of life. This process begins with gene expression to RNA. Insights into the inherent variance in gene expression that underlies different phenotypes (and thereby the cis-regulatory logic) could be gained through predicting the strength of gene expression from DNA sequence. Big data provides a hypothesis-free approach to model the promoter expression strength from regulatory sequences. Open innovation challenges on problems involving big data provide a framework for independent verification of the effectiveness of multiple computing methods on common benchmark data sets. Here, we discuss our participation in the DREAM Challenge 2022, on the problem of promoter strength prediction from sequence. Our efforts include the development of an attention-based ConvLSTM-BiLSTM model with sample weighting that appears to achieve Spearman rank correlation >0.9 on the designated test set.

**1. Description of data usage**

The length of the regulatory sequence to be considered was set to 110 core nucleotides. Longer regulatory sequences were truncated equally at the 5' and 3' ends to achieve this length. Shorter regulatory sequences were padded with ‘N’ equally on both ends to achieve this length. Parity in length was accounted. We preprocessed this dataset by applying one hot encoding on the alphabet: ‘ATGCN’. This yielded the final preprocessed training data of dimensions (6739258, 110, 5). The sequences in test data were similarly preprocessed to yield 110-nucleotide one-hot encoded vectors. In addition, we inspected the expression data, and noticed that it was approximately Gaussian with a heavy left skew. In order to be able to predict the outlier expression values effectively, the instances in the left tail may need to be over-sampled/over-weighted. This was done by standardizing the expression values and dropping the sign of the z-score. The train data was saved as a binary file and loaded at the time of model training.

**2. Description of the model**

We designed a ConvLSTM - BiLSTM model with one attention layer as our first steps to the problem. BiLSTM provides reading in both directions to acount for the effects of promoter regions in the reverse strand. The idea was implemented using a standard Sequential model with the following layers in order:

1. ConvLSTM1D layer with 32 filters and kernel\_size=3

2. Another ConvLSTM1D layer identical in configuration to the first layer

3. Batch Normalization, relu activation, dropout =0.15 followed by MaxPooling2D with kernel = (2,2)

4. A third ConvLSTM1D layer with 64 filters and kernel\_size=5

5. Batch Normalization, relu activation, dropout =0.15 followed by MaxPooling2D with kernel = (2,2)

6. Bidirectional LSTM with 128 nodes, followed by dropout=0.1

7. Bidirectional LSTM with 32 nodes, followed by dropout=0.1

8. A generic attention layer

9. Gentle fanning out into a Dense layer with 128 units, followed by Batch Normalization, gelu activation and Dropout = 0.1

10. Dense layer with 32 units, followed by Batch Normalization, gelu activation and Dropout = 0.1

11. Output unit.

**3. Training procedure**

A few loss functions were tried during model training. Typically we used the correlation loss as the objective function and one of the other functions (listed below) as the validation loss to be monitored for early stopping.

(1) Correlation loss

(2) Huber loss

(3) MAE (Mean absolute error) - this was preferred instead of MSE since there were outlier expression values in the training data.

Since we did not find much use in early stopping, we settled on correlation loss as the objective function and trained the model on the full data for five epochs. In addition we used a custom activation function, 'gelu' - Gaussian error linear units, for the terminal Dense layers [2]. The adam optimizer with learning rat = 3E-4 was used for the gradient descent. Regularization was performed for all weights (kernel and bias) in the ConvLSTM and Dense layers using L2. The same weights were initialized using HeNormal (for the kernel ones) and Random Uniform (for the bias). This model was used to generate the predictions on the test dataset.

**4. Other important features**

[Please describe any other information on your model that you feel is important and was not included in the above sections.]

**5. Contributions and Acknowledgement**

**5.1 Contributions**

| **Name** | **Affiliation Email** |
| --- | --- |
| Ashok Palaniappan | Associate Professor, School of Chemical and Biotechnology, SASTRA University.  Email: apalania@scbt.sastra.ac.in; ashokpalaniappan78@gmail.com  Design; preprocessing; model building; model evaluation; writing - draft & revision. |
| Krishnakant Gupta | Research Scientist, School of Chemical and Biotechnology, SASTRA University.  Email:krishnakant@scbt.sastra.ac.in  Preprocessing; model building; writing - draft. |
| Prasanna Kumar S | Undergraduate Student, SASTRA University.  Email: sprasanna353@gmail.com  Preprocessing; model building; model evaluation; writing - draft. |

**5.2 Acknowledgement**

We would like to thank the School of Chemical and Biotechnology, SASTRA deemed University for infrastructure and computing support.

**6. References**

1. Vaishnav ED, de Boer CG, Molinet J, Yassour M, Fan L, Adiconis X, Thompson DA, Levin JZ, Cubillos FA, Regev A. The evolution, evolvability and engineering of gene regulatory DNA. Nature. 2022 Mar;603(7901):455-463. doi: 10.1038/s41586-022-04506-6. Epub 2022 Mar 9. PMID: 35264797; PMCID: PMC8934302.
2. Avsec, Ž., Agarwal, V., Visentin, D. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat Methods* 18, 1196–1203 (2021). https://doi.org/10.1038/s41592-021-01252-x

.

**7. Feedback**

This competition was truly an adventure to us. We got to learn new things via participating in this competition, especially in:

1. building end-to-end sophisticated deep learning models
2. working in the cloud, with Virtual Machines equipped with TPUs graciously provided by Google Cloud Computing resources in conjunction with the DREAM challenge consortium

We would like to thank the competition Committee for hosting this fabulous event and sharing their data and models in the first place.