Deep Learning-Based Differential Gene Expression Prediction of Ischemic Stroke among Sickle Cell Anemia Patients

Zhongyi (James) Guo (SUID: guozy), Tiankai Yan (SUID: ytk2304), Luna Lyu (SUID: lunalyu)

Introduction

Sickle cell anemia (SCA) is a genetic hematological disorder in which malfunctioning hemoglobin can lead to lethal conditions, such as ischemic stroke (IS) where blood clots block blood flow to the brain (Ware et al., 2017). According to the National Heart, Lung, and Blood Institute (NHLBI, as of September 2024, it affects over 100,000 people in the United States and 8 million people globally (NHLBI, 2024).

Many studies investigated differential gene expression related to IS (Appunni et al., 2020; Rust, 2023), but few focused specifically on SCA patients. Additionally, many studies relied on traditional biological and statistical methods to analyze gene expression in strokes among SCA patients (Ito et al., 2020; Theofilatos et al., 2019), but they focused on the genes themselves rather than the regulatory regions that drived differential gene expression.

In this study, we applied deep learning and statistical methods to investigate gene expression regulation patterns associated with IS among SCA patients, using upstream putative regulatory promoters that initiate gene transcriptions. Variations in promoter sequences can impact the binding affinity of transcription factors and RNA polymerase, thereby modulating transcription. Our approaches integrate biological, computational, and statistical insights, extending the analysis beyond traditional methods.

Our project holds promising applications, providing evidence for (1) differentially expressed genes associated with IS among SCA patients, (2) the design of synthetic genomic sequences for gene editing and other therapeutic interventions, and (3) predictive modeling and risk assessment to evaluate IS risk among SCA patients, guiding preventative healthcare.

Data

Source Data

We used bulk RNA-seq count data retrieved from the Gene Expression Omnibus (GEO) database (GSE248760). This case-control study included 8 participants: 4 SCA patients with IS and 4 controls, all recruited in São Paulo, Brazil. Peripheral blood samples containing circulating endothelial colony-forming cells were collected from these participants and assayed for RNA read counts using the Illumina HiSeq 2500 platform. The measured count matrix included 58,174 genes, each labeled with an Ensembl Gene Identifier, documenting read counts for each gene across all samples.

Differential Gene Expression (DGE) Analysis

We performed DGE analysis with *DESeq2* in R, using a threshold of log2 fold change > |2| and FDR-adjusted p-value < 0.05. Genes meeting these criteria were labeled as "differentially expressed" (1), while others were labeled as not differentially expressed (0), providing a binary ground truth for subsequent modeling.

Upstream Promoter Sequence Extraction

Promoters that initiate gene expression are located 2,000 base pairs (bp) upstream the transcription start site (TSS) (Kim et al., 2020). We used *biomaRt* in R to retrieve genomic sequences located 2,000 bp upstream of TSS for all genes, based on the *Human genes (GRCh38.p14)* dataset from *Ensembl*. The extracted DNA sequences were one-hot encoded (A = [1,0,0,0], T = [0,1,0,0], C = [0,0,1,0], G = [0,0,0,1]).

2D Convolutional Neural Network (CNN)

We used a mini-batch size of 256, trained for 20 epochs, shuffled the data with a train/dev split of 80/20, and trained the 2D CNN using Tensorflow. The CNN architecture follows a VGG-like structure, starting with a 2D convolutional layer with 16 filters of size (2, 2), using ReLU activation and same padding to maintain dimensions. This is followed by a max pooling layer, and then a second convolutional layer with 32 filters, also followed by max pooling. After these layers, a global max pooling layer condenses the feature maps, feeding into a dense layer with 64 units and ReLU activation. The model concludes with a sigmoid output layer for binary classification of differential expression.

We used the Adam optimizer with a default learning rate of 0.001, binary cross-entropy as the loss function, and accuracy as the evaluation metric to train the model. The accuracy and loss for both training and dev sets were visualized.

1D *Convolutional Neural Network + LSTM*

We used PyTorch to train the model using a mini-batch size of 256 over 5 epochs. The model starts with a 1D convolutional layer with 64 filters of size 3, ReLU activation, batch normalization, and max pooling with a pool size of 2. A second convolutional layer with 128 filters follows, also with ReLU activation, batch normalization, and max pooling. Sequential outputs are passed to a bidirectional LSTM with two layers, each with a hidden size of 64, and a dropout rate of 0.3 to capture long-range dependencies. The LSTM output is mean-pooled across time and fed into a dense layer with 64 units, ReLU activation, batch normalization, and dropout. A final dense layer with two outputs provides classification scores. The model is trained using the Adam optimizer, cross-entropy loss, and gradient clipping (max norm of 1.0). A scheduler reduces the learning rate if validation accuracy stagnates. Training and validation performance are tracked and visualized across epochs.

Transformer

We used PyTorch to train the model using a mini-batch size of 256 over 5 epochs. A linear embedding layer maps 4-dimensional inputs to a 128-dimensional space, with learnable positional encodings added to incorporate positional information. The input is passed through a 3-layer transformer encoder with 8 attention heads and feedforward layers of size 256, using dropout (0.1) for regularization. The transformer output is mean-pooled across the sequence dimension to generate a fixed-size representation. This representation is passed through a fully connected layer with 64 units, ReLU activation, and dropout, followed by a final layer that outputs logits for binary classification. The model is trained with the AdamW optimizer, cosine annealing for learning rate scheduling, and cross-entropy loss. Gradient clipping (max norm 1.0) ensures training stability, with loss and accuracy tracked for both training and validation phases.

Generative Adversarial Network (GAN)

Due to label imbalance, with many more genes labeled as not differentially expressed than as differentially expressed, classification bias was observed in the CNN model. To address this, we will train a GAN to generate high-quality promoter sequences to achieve balanced labels, aiming to mitigate the classification bias in the CNN.

The generator, modeled after a VGG-like architecture but customized for gene sequence analysis, effectively produced synthetic samples that closely resembled real gene sequences. The discriminator used a simpler convolutional model and non-saturating loss.

However, the generator provides the continuous value instead of a binary label in the data. Binarizing generated data might lead to mode collapse. We will devise appropriate strategies to address this situation should it arise.

Preliminary Results

DGE Analysis

We identified 4,126 genes as statistically significant differentially expressed (1), while 51,097 genes were not (0). Only 7.1% of genes were labeled as 1, leading to classification bias in the CNN model we trained due to imbalanced labels.

CNN

According to Figure 1, the crude CNN converged at the first epoch, as the training loss reached an elbow point and gradually decreased to approximately 0.265 over subsequent epochs, while the dev loss showed a similar decreasing trend, stabilizing around 0.252. Meanwhile, training and dev accuracy remained stable from the first epoch, both hovering around 0.925. This indicates that our CNN requires fine-tuning to further reduce training and also dev loss.

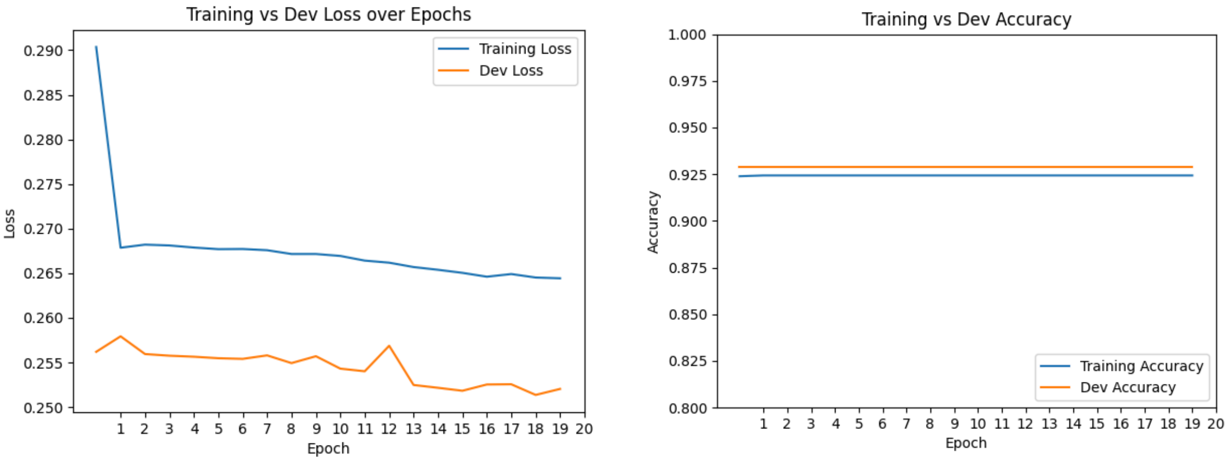

Figure 1: Training vs Dev Loss and Accuracy (2D CNN)

1D CNN + LSTM & Transformer:

We have established the models. The results are not yet available.

Discussion

Next Step

The next steps for our project are to (1) fine tune the 2D CNN, 1D CNN + LSTM, and the transformer to achieve the optimal performance, (2) continue GAN training to achieve a better performance of the generator, (3) apply the generated sequences to 2D CNN, 1D CNN + LSTM, and the transformer, and examine if they will improve the performance.

Compared to the GAN model, the diffusion models might have advantages in model stability. Unlike GANs, diffusion models generate data through an iterative denoising process, which could be better suited for binary labels in cells. This method would allow the generator to produce synthetic samples gradually, maintaining the discrete (0 or 1) nature of gene sequences without post-processing binarization. The D part in GAN could also be used to compute the label for the generated gene sequence

Code Availability

You can access our GitHub repository here: https://github.com/JG1ANDONLY/SCA-DL-DGE

References

Ware, R. E., de Montalembert, M., Tshilolo, L., & Abboud, M. R. (2017). Sickle cell disease. *The Lancet*, *390*(10091), 311-323.

Sickle Cell Disease - What Is Sickle Cell Disease? (2024, September 30). NHLBI. Retrieved November 14, 2024, from https://www.nhlbi.nih.gov/health/sickle-cell-disease

Appunni, S., Rubens, M., Ramamoorthy, V., Sharma, H., Singh, A. K., Swarup, V., & Singh, H. N. (2020). Differentially expressed genes and their clinical significance in ischaemic stroke: An in-silico study. *The Malaysian Journal of Medical Sciences: MJMS*, *27*(6), 53.

Rust, R. (2023). Ischemic stroke-related gene expression profiles across species: a meta-analysis. *Journal of Inflammation*, *20*(1), 21.

Ito, M. T., da Silva Costa, S. M., Baptista, L. C., Carvalho‐Siqueira, G. Q., Albuquerque, D. M., Rios, V. M., ... & Melo, M. B. (2020). Angiogenesis‐related genes in endothelial progenitor cells may be involved in sickle cell stroke. *Journal of the American Heart Association*, *9*(3), e014143.

Theofilatos, K., Korfiati, A., Mavroudi, S., Cowperthwaite, M. C., & Shpak, M. (2019). Discovery of stroke-related blood biomarkers from gene expression network models. *BMC medical genomics*, *12*, 1-15.

Kim, S., Park, H. J., Cui, X., & Zhi, D. (2020). Collective effects of long-range DNA methylations predict gene expressions and estimate phenotypes in cancer. *Scientific reports*, *10*(1), 3920.