CS230

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

Zhongyi Guo
Stanford University
guozy@stanford.edu

Luna Lyu
Stanford University
lunalyu@stanford.edu

Tiankai Yan
Stanford University
ytk2304@stanford.edu

Abstract

We combined statistical differential gene expression analysis with advanced deep learning models, including Convolutional Neural Networks, and XGBoost, to classify differential gene expression from 2000 bp upstream putative promoter sequences. This approach explores a relatively underexamined area, offering novel insights into the connection between promoter sequences and gene expression. A major challenge was the scarcity and imbalance of relevant data. To overcome this, we implemented generative model, diffusion, to augment the dataset by generating synthetic promoter sequences.

1 Introduction

Sickle cell anemia (SCA) is a genetic hematologic disorder in which malfunctioning hemoglobin can lead to lethal conditions, such as ischemic stroke (IS), in which blood clots block blood flow to the brain [17]. According to the National Heart, Lung and Blood Institute (NHLBI), it affects more than 100,000 people in the United States and 8 million people worldwide [12].

In this study, we applied supervised and unsupervised deep learning and statistical methods to investigate the gene expression regulation patterns associated with IS among patients with SCA, using upstream putative regulatory promoters that initiate gene transcriptions. Variations in promoter sequences could affect the binding affinity of transcription factors and RNA polymerase, thereby modulating transcription (Figure 1). Our approaches integrated biological, computational, and statistical insights, extending the analysis beyond traditional methods. We combined statistical differential gene expression analysis with advanced deep learning models, including Convolutional Neural Networks, XGBoost, and LSTM, to classify differential gene expression from 2000 bp upstream putative promoter sequences. This approach explores a relatively underexamined area, offering novel insights into the connection between promoter sequences and gene expression.

The input for our deep learning models consisted of upstream putative promoter sequences for each gene, each 2000 base pairs in length and composed of adenine (A), thymine (T), cytosine (C), and guanine (G). These sequences were one-hot-encoded, with A = [1,0,0,0], C = [0,1,0,0], G = [0,0,1,0], and T = [0,0,0,1]. Each sequence was labeled to indicate whether the immediately downstream gene was differentially expressed between SCA patients with IS and those without IS. Using a one-hot-encoded 2000bp genomic sequence of A, T, C, and G, the models produced a binary output, predicting whether its immediate downstream gene would be differentially expressed or not.

Our project held promising applications, providing evidence for (1) differentially expressed genes (DEGs) 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.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

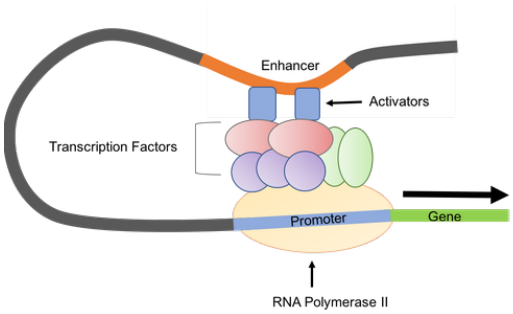

Figure 1: Transcription Factor Binding Dynamics with Promoters and Enhancers

2 Related work

Many studies investigated differential gene expression (DGE) related to IS [3][13], but few focused specifically on patients with SCA. Furthermore, many studies relied on traditional biological and statistical methods to analyze gene expression in strokes among SCA patients [8][16], but they focused on the genes themselves rather than the regulatory regions that drive DGE.

2.1 DGE Analysis for IS among patients with SCA

Ito et al. used DGE analysis to identify genes associated with stroke versus without stroke among SCA patients [8]. However, their study was too broad, as the three stroke syndromes mentioned in their paper—clinical ischemic stroke, hemorrhagic stroke, and clinically silent stroke—could all arise from multifocal small vessel disease yet differ in their specific pathogenesis. For example, ischemic stroke resulted from sickled red blood cells adhering to the endothelium and causing vascular occlusion, whereas hemorrhagic stroke occurred due to blood vessel rupture.

In the paper from which this dataset originated [6], the authors performed DGE analysis using limma. limma applied stricter FDR control compared to DESeq2, resulting in fewer identified DEG.

2.2 Deep Learning for Genomics

Numerous researchers have highlighted the revolutionary power of deep learning in genomics research. Zou et al. presented a workflow for training convolutional and recurrent neural networks on DNA sequences, followed by evaluating and interpreting the results [19]. We adapted this workflow for our study without a validation set due to the smaller sample size than typical deep learning datasets.

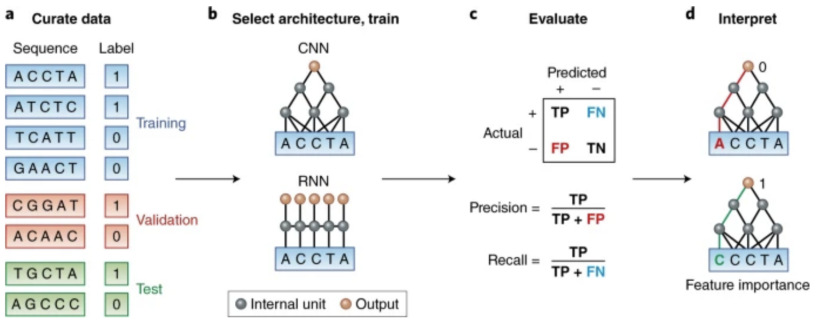

Figure 2: Workflow for Deep Learning Applications in Genomics

Moreover, many researchers have developed deep learning architectures for genomic sequences. For example, *BPNet* was developed to analyze motifs and the syntax of *cis*-regulatory sequences in genomics, leveraging convolutional neural networks (CNNs) to study cooperative transcription factor binding interactions [4]. *DeepChrome* was designed to predict gene expression by modeling combinatorial interactions with histone modification marks in a cell-type-specific level using a

deep CNN [15]. *DeepSEA* deciphered regulatory sequence patterns from large-scale chromatin profiling data, allowing for the prediction of chromatin changes caused by sequence alterations with single-nucleotide resolution using a deep CNN [18]. All of them used CNNs to model genomic and epigenomic sequences, highlighting the transformative potential of deep learning, especially CNN, in genomic research. They groundbreakingly addressed various aspects of genomic research, including genomic sequences, chromatin accessibility, and single nucleotides. Recognizing the potential of deep learning in genomic research, we adopted the workflow developed by Zou et al. to integrate statistical differential expression analysis with some deep learning models in our study. **The uniqueness and novelty of our project were that we used a component of the gene regulatory network, leveraging regulatory genomic sequence syntax besides chromatin accessibility.**

3 Dataset and Features

3.1 Source Data

We used raw bulk RNA-seq count data retrieved from the Gene Expression Omnibus database (GSE248760) [6]. This case-control study included 8 participants: 4 SCA patients with IS and 4 without IS, all recruited in São Paulo, Brazil. Peripheral blood samples containing circulating endothelial colony-forming cells were collected 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 across all genes and all samples.

3.2 DGE Analysis

We performed DGE analysis with DESeq2 in R [11], 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. Of the 58,174 genes in the dataset, 4,324 were identified as differentially expressed (labeled as 1, positive) and 53,850 were not differentially expressed (labeled as 0, negative).

3.3 Upstream Promoter Sequence Extraction

Promoters that initiated gene expression were located 2,000 base pairs (bp) upstream the transcription start site (TSS) [10]. 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].).

3.4 Model Training Dataset

The final comprehensive dataset for model training had dimensions of 58,174 × 3, comprising gene IDs, labels, and one-hot encoded sequences, with each sequence having dimensions of 2000 × 4.

We shuffled the data and randomly allocated 80% to the training set and 20% to the test set, resulting in 44,178 one-hot-encoded promoter sequences with labels for training and 11,045 for testing before data augmentation and downsampling.

4 Methods

In our project, we adopted some classification models to address classification task for gene expression regulation patterns associated with IS among SCA patients. However, due to a severe label imbalance in the dataset, with 4,324 positive sequences (7.4%) and 53,850 negative ones (92.6%), we developed generative models to generate positive sequences. These generated sequences were incorporated into the classification model to address the imbalance. Moreover, we experimented with downsampling by randomly selecting a subset of positive samples to match the number of negative samples.

4.1 Baseline CNN

CNN offers advantages for our task by targeting specific regions in promoter sequences where transcription factors bind and leveraging convolution filters for similar patterns across promoter

sequences. We used a VGG-like architecture [14] with blocks of 2D-convolution, batch normalization, and max pooling layers, followed by dense layers with ReLU activation and a final sigmoid layer for binary output. Binary cross-entropy was chosen as the loss function with the Adam optimizer (learning rate 0.001) in TensorFlow [2] for the baseline CNN model (Figure 4).

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

Figure 3: Binary Loss Entropy - Loss Function

4.2 CNN + XGBoost

XGBoost boosts a strong learner while adopting various techniques against overfitting, achieving an optimal balance in the bias-various trade off [5]. However, flattening the input results in an excess number of features for XGBoost. To address this concern, we adopt a hybrid CNN + XGBoost architecture. The CNN follows the baseline CNN structure, but with a tree-based XGBoost classifier. The integrated architecture achieves the balance in bias-variance trade-off and maintains computational efficiency.

XGB follows the hyper-parameters tuning: 1) fix learning rate and number of rounds, 2) tune max depth of tree pruning and min child weight of spliting, 3) tune gamma, 4) tune subsample and colsample, 5) tune alpha and lambda in regularization term, 6) tune learning rate & number of rounds.

4.3 Transformer

This transformer-based model processes sequential DNA data using self-attention to capture local and global dependencies. An embedding layer projects inputs into a higher-dimensional space, adding positional encodings to preserve sequence order. The model includes a transformer encoder with 3 layers, 8 attention heads. Globally pooled features are passed through a fully connected layer (64 units, ReLU, dropout) and a final dense layer for binary classification. Gradient clipping (max norm 1.0), a cosine annealing scheduler and weighted cross-entropy loss addresses class imbalance are also utilized in transformer.

4.4 Diffusion Model

The diffusion model learns to denoise input promoter sequences progressively, using time embedding module to capture temporal dynamics [7]. The architecture starts with a convolutional layer projecting 4-channel DNA sequences into a 128-dimensional space, followed by residual convolutional blocks with group normalization and SiLU activations that integrate time embeddings at each block. A beta regulates the process of adding noise and subsequently denoising during model training, where the model minimizes mean squared error between predicted and actual noise to reverse the diffusion process and capture intricate genetic patterns effectively. Only data labeled as 1 were used to train the diffusion model, and the generated sequences were used as data augmentation to alleviate the label imbalance issue when training other models.

5 Experiments/Results/Discussion

We used a mini-batch size of 256, and epoch size of 20 to train cnn and transformers. They used the Adam optimizer with learning rate of 1e-3. The hyper-parameters for XGBoost included learning rate, number of rounds, max depth, min child weight, gamma, subsample, colsample, alpha and lambda. They were tuned after down sampling: 0.01, 100, 7, 2, 0.167, 0.8, 0.6, 0.5, 1, correspondingly.

5.1 Classifier

The following tables present the classification result for the data before the down sampling and after the down sampling.

The CNN + XGBoost model achieved an AUC of 0.56 before downsampling, which improved to 0.79 after downsampling.

Metric	CNN	CNN+XGB	Transformer
Accuracy	0.93	0.61	0.71
Precision	0.0	0.18	0.11
Recall	0.0	0.28	0.42
F1-Score	0.0	0.22	0.17

Table 1: Performance Metrics

Metric	CNN+XGB	Transformer
Accuracy	0.68	0.57
Precision	0.68	0.59
Recall	0.73	0.40
F1-Score	0.70	0.48

Table 2: Performance Metrics after Down Sampling

5.2 Generators

We implemented two generators, GAN and diffusion, and based on prior experiments and observations, diffusion was found to outperform GAN. Using the diffusion model, we generated 55,000 synthetic positive promoter sequences. Incorporating these sequences into the transformer model resulted in an accuracy of 56.81%, precision of 0.59, recall of 0.40, and an F1-score of 0.48. However, the limited diversity in the positive samples may have constrained the model's learning capacity on regonize diversity.

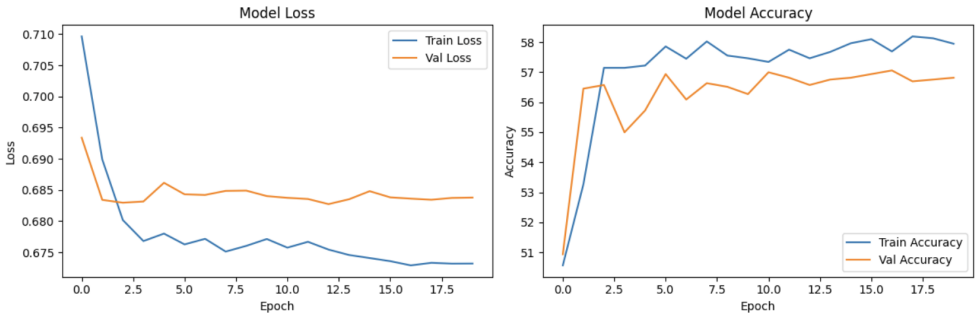

Figure 4: Transformer Performance after Incorporating Synthetic Sequences from Diffusion Model

Actual/Predicted	Positive	Negative
Positive	325	485
Negative	228	613

Table 3: Confusion Matrix of Balanced Transformer

5.3 Evaluation

The CNN+XGB and transformer models, after down-sampling, both demonstrate better classification performance. However, hyperparameter tuning for XGB requires significant computational resources, while transformers are less affected by this issue. Conversely, transformers demand longer time for each iteraton compared to XGB, making their implementation also resource-intensive.

Additionally, feature extraction approach lacks interpretability, and while down-sampling improves results, it compromises generalizability. Augmenting the dataset with generated positive samples could effectively address these challenges by enhancing diversity and mitigating the limitations of down-sampling.

6 Conclusion/Future Work

Initially, we aimed to integrate enhancer sequences alongside promoter sequences. However, we were unable to find reliable databases for circulating endothelial colony-forming cells. In future work, we plan to include enhancer sequences to enhance the analysis.

Additionally, we could utilize a more comprehensive model, such as CvT [9], (Convolutional Vision Transformers), or a model having better balance between performance and efficiency, such as LightGBM [1].

7 Contributions

We also tried our models, including GAN, 1DCNN + LSTM, etc.

Luna: Designed, implemented, and trained 1D Conv + LSTM, Transformer, GAN, and Diffusion.

Zhongyi (James): Performed differential gene expression, extracted promoter sequences, one-hot encoded them, and trained 2D Conv CNN. Experimented GAN but its performance was inferior to Luna's and Tiankai's models.

Tiankai Yan: Designed, implemented, and trained CNN, GAN, Diffusion, and CNN+XGB. Conducted hyperparameters tuning.

8 Codes

You can access our repo through https://github.com/JG1ANDONLY/SCA-DL-DGE

References

[1] Lightgbm: A highly efficient gradient boosting decision tree. *In Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Sandeep Appunni, Muni Rubens, Venkataraghavan Ramamoorthy, Hina Sharma, Anjani Kumar Singh, Vishnu Swarup, and Himanshu Narayan Singh. Differentially expressed genes and their clinical significance in ischaemic stroke: An in-silico study. *The Malaysian Journal of Medical Sciences: MJMS*, 27(6):53, 2020.

[4] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature genetics*, 53(3):354–366, 2021.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] Júlia Nicoliello Pereira de Castro, Sueli Matilde da Silva Costa, Ana Carolina Lima Camargo, Mirta Tomie Ito, Bruno Batista de Souza, Victor de Haidar e Bertozzo, Thiago Adalton Rosa Rodrigues, Carolina Lanaro, Dulcinéia Martins de Albuquerque, Roberta Casagrande Saez, et al. Comparative transcriptomic analysis of circulating endothelial cells in sickle cell stroke. *Annals of Hematology*, 103(4):1167–1179, 2024.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[8] Mirta T Ito, Sueli M da Silva Costa, Letícia C Baptista, Gabriela Q Carvalho-Siqueira, Dulcinéia M Albuquerque, Vinicius M Rios, Stephanie Ospina-Prieto, Roberta C Saez, Karla P Vieira, Fernando Cendes, et al. Angiogenesis-related genes in endothelial progenitor cells may be involved in sickle cell stroke. *Journal of the American Heart Association*, 9(3):e014143, 2020.

[9] Meng Q. Finley T. Wang T. Chen W. Ma W. ... Liu T. Y. Ke, G. Cvt: Introducing convolutions to vision transformers. *Advances in neural information processing systems*, 2017.

[10] Soyeon Kim, Hyun Jung Park, Xiangqin Cui, and Degui Zhi. Collective effects of long-range dna methylations predict gene expressions and estimate phenotypes in cancer. *Scientific reports*, 10(1):3920, 2020.

[11] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15:1–21, 2014.

[12] National Heart, Lung, and Blood Institute (NHLBI). Sickle cell disease. `https://www.nhlbi.nih.gov/health/sickle-cell-disease`, 2024. Accessed: 2024-11-30.

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

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 2016.

[16] Konstantinos Theofilatos, Aigli Korfiati, Seferina Mavroudi, Matthew C Cowperthwaite, and Max Shpak. Discovery of stroke-related blood biomarkers from gene expression network models. *BMC medical genomics*, 12:1–15, 2019.

[17] Russell E Ware, Mariane de Montalembert, Léon Tshilolo, and Miguel R Abboud. Sickle cell disease. *The Lancet*, 390(10091):311–323, 2017.

[18] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[19] James Zou, Mikael Huss, Abubakar Abid, Pejman Mohammadi, Ali Torkamani, and Amalio Telenti. A primer on deep learning in genomics. *Nature genetics*, 51(1):12–18, 2019.