DATASCI223 Final Project

LADTransformer: Predicting Lamina-Associated Domains in DNA Sequences

1. Introduction

1.1 Problem Statement

The goal of this project is to develop a deep learning model called LADTransformer, that can predict whether a provided DNA sequence is in a lamina-associated domain (LAD), genomic regions tethered to the nuclear lamina, a nuclear compartment that plays a role in 3D genome organization and gene expression.

1.2 Motivation and Inspiration

Lamina-associated domains (LADs) are genomic regions that interact with the nuclear lamina, a meshwork of proteins adjacent to the nuclear lamina that forms a generally transcriptionally repressive compartment. LADs are thought to play a crucial role in genome organization and regulation, and their architecture and dynamics have been shown to vary across different cell types, developmental stages, and disease states. Recent studies have revealed that LADs in the human forebrain are enriched for transcriptionally active neural genes associated with synapse function, highlighting their functional significance. Understanding the relationship between DNA sequences and LAD formation is a key part of deciphering the regulatory logic of non-coding DNA. DNA is comprised of two types of sequence: 1) protein-coding sequence, which is universal, and 2) regulatory sequence, which varies across organisms, cell types, and developmental stages.

Non-coding DNA is highly similar to natural language. For example, identical regulatory elements can have different functions in different contexts, suggesting regulatory sequence may be polysemantic. In addition, long-range interactions between distant parts of the genome suggest distant semantic relationships. These are both core properties of natural language. This similarity makes the transformer deep learning architecture an ideal approach. Transformers can capture long-range dependencies and interactions between distant positions in the DNA sequence, which is crucial for understanding regulatory mechanisms.

DNABERT and EpiGePT are two recent transformer-based models that have shown promising results in understanding the epigenomic language and predicting epigenomic signals from DNA sequences. DNABERT is a pre-trained bidirectional encoder representation that captures global and transferable understanding of genomic DNA sequences. It has been successfully applied to predict promoters, splice sites, and transcription factor binding sites after fine-tuning with small task-specific labeled data. DNABERT's ability to visualize nucleotide-level importance and semantic relationships within input sequences has provided valuable insights into conserved sequence motifs and functional genetic variant candidates. EpiGePT is a transformer-based pre-trained language model that predicts context-specific epigenomic signals and chromatin contacts by incorporating context-specific activities of transcription factors and 3D genome interactions. EpiGePT's context-dependent input and output enable it to predict epigenomic signals and chromatin interactions in new cellular contexts, offering wider applicability and deeper biological insights compared to models trained solely on DNA sequences. By developing a model that can accurately predict LAD regions from DNA sequences, we can gain valuable insights into the syntax, grammar, and semantics within regulatory DNA sequences.

2. Dataset

2.1 Description

The dataset used in this project consists of approximately 1.2 million examples of DNA sequence from the reference human genome, each spanning 20,000 base pairs (20kb). Each 20kb sequence is assigned a continuous percentage, which represents the proportion of the sequence which was found to be associated with nuclear lamina. The LAD data was collected using Genome Organization using CUT and RUN Technology (GO-CaRT), a method used to map genomic interactions with the nuclear lamina and identify lamina-associated domains (LADs). GO-CaRT uses a LaminB1 antibody to recruit protein A micrococcal nuclease (pA-MNase) to the nuclear lamina, resulting in targeted cleavage of lamina-proximal DNA which is then sequenced to identify LAD regions. The DNA sequences are stored in .npz files, with each file containing two arrays: sequences, and lad_percentages. The sequences array represents the DNA sequences, and the lad_percentages array contains the corresponding LAD percentages for each sequence.

2.2 Features and Outcome

The input featurs for the model are the 20kb DNA sequences, which have been integer encoded (in the data_preprocessing.py file). The mapping is as follows: ‘A’ = 0, ‘C’ = 1, ‘G’ = 2, ‘T’ = 3, ‘N’ = 4. The outcome is the continuous LAD percentage.

3. Code and Dependencies

3.1 Code Overview

The code for this project is written in Python and utilizes various libraries and frameworks for data processing, model training, and evaluation. The main components of the code include:

- Data loading and preprocessing functions
- LAD Transformer model definition
- Training and validation loop
- Evaluation on the test set
- Experiment tracking using Weights and Biases (wandb)

3.2 Dependencies

To run the code, the following dependencies are required:

- Python 3.x
- Bio
- wandb
- numpy
- pandas
- scikit-learn
- torch

- matplotlib
- tqdm
- tensorflow

3.3 How to Run the Code

1. Clone the repository and navigate to the project directory.
2. Install the required dependencies using `pip install Bio wandb numpy pandas scikit-learn torch matplotlib tqdm tensorflow`.
3. Ensure that the encoded sequence data is stored in a Google Drive directory named "encoded_sequences".
4. Mount your Google Drive in the Colab notebook or the environment where you will be running the code.
5. Log in to Weights and Biases (wandb) for experiment tracking using `wandb.login()`.
6. Set the desired hyperparameters in the code.
7. Run the code in a Colab notebook or a Python environment.

4. Decisions and Trade-Offs

During the development of this project, several decisions were made to balance performance, computational efficiency, and time constraints. Some of the key decisions and trade-offs include:

- Using a smaller batch size (`chunk_size`) to reduce memory usage, which may impact training speed.
- Limiting the validation subset size to a fraction of the total validation data to speed up the validation process, potentially affecting the accuracy of validation metrics.
- Choosing specific hyperparameters, such as the number of convolutional and transformer layers, hidden sizes, and learning rate, based on experimentation and available computational resources.

Three convolutional and three max pooling layers were selected so the length of the input DNA sequence could be reduced from 20kb to 2.5kb. The final hyperparameter configuration for the model was:

- Number of convolutional layers: 3
- Convolution hidden size: 32
- Number of transformer layers: 4
- Transformer hidden size: 128
- Number of attention heads: 4
- Dropout: 0.1
- Learning rate: 0.001
- Number of classes: 5 (for ‘A’, ‘T’, ‘G’, ‘C’, and ‘N’)
- Chunk size: 100

The model was trained with access to one A100 GPU.

5. Example Output

The LADTransformer model takes integer-encoded DNA sequences as input and predicts the corresponding LAD percentages. During training, the model logs the training and validation losses at each step using wandb. After training, the model is evaluated on the test set, and the test loss and mean squared error (MSE) between the predicted and actual LAD percentages are calculated and printed in the console.

6. Results

The LAD Transformer model was trained on the DNA sequence and LAD percentage dataset using the specified hyperparameters and architecture. The training process was monitored using the training and validation loss, which were logged at regular intervals using Weights and Biases (wandb) for visualization and tracking.

After training, the model was evaluated on the test set to assess its performance on unseen data. The test loss, which measures the discrepancy between the predicted and actual LAD percentages, was found to be 0.2504. Additionally, the mean squared error (MSE) between the predicted and actual LAD percentages on the test set was calculated to be 0.2504.

```
Test Loss: 0.2504, Test MSE: 0.2504
```

The moderately low test loss and MSE values suggest that the LAD Transformer model may be able to learn meaningful patterns and relationships between the DNA sequences and LAD percentages, but increasing the size and the complexity of the model is needed to improve it’s performance.

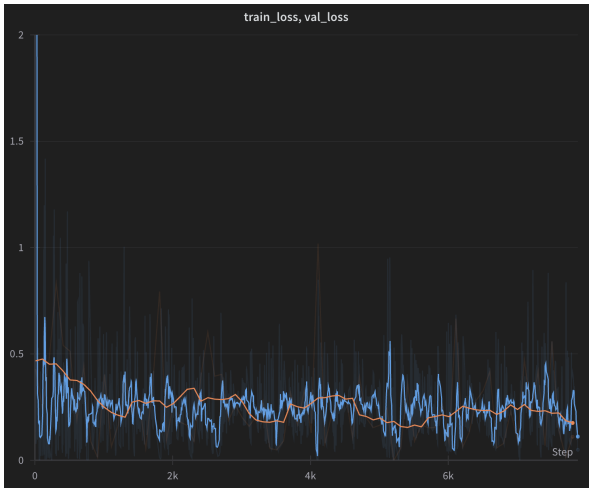

Figure 1. Training and validation MSE loss curves for the LADTransformer model over 8,000 steps.

The plot (Figure 1) shows the training and validation loss values over the course of training the LAD Transformer model. The training loss (blue line) represents the model's performance on the training dataset, while the validation loss (orange line) indicates the model's performance on the held-out validation dataset. The x-axis represents the training steps, and the y-axis represents the loss values.

References

Ahanger, S. H., Delgado, R. N., Gil, E., Cole, M. A., Zhao, J., Hong, S. J., Kriegstein, A. R., Nowakowski, T. J., Pollen, A. A., & Lim, D. A. (2021). Distinct nuclear compartment-associated genome architecture in the developing mammalian brain. *Nature Neuroscience, 24*(9), 1235-1242.

Avsec, Z., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., & Kelley, D. R. (2021). Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods, 18*(10), 1196-1203.

Gao, Z., Liu, Q., Zeng, W., Jiang, R., & Wong, W. H. (2023). EpiGePT: a pretrained transformer model for epigenomics. *bioRxiv preprint*.

Ji, Y., Zhou, Z., Liu, H., & Davuluri, R. V. (2021). DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics, 37*(15), 2112-2120.