DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences

Team DanQ: Phila Dlamini, Jesalina Phan, Rosy Sun, Megan Carlson

Introduction

Paper: DanQ, Daniel Quang et al. 2016

Problem

- Build a model that predicts the function of non-coding DNA directly from sequence

What's novel

- Previous approaches relied solely on CNNs (e.g., DeepSEA), which are unable to capture long-range dependencies, or on SVMs (e.g., gkm-SVM), which require extensive feature engineering.
- DanQ uses CNNs to detect local motifs, and BiLSTMs to model the regulatory grammar between them

Why it's important

- Over 98% of the genome is non-coding, yet ~93% of disease-linked genetic variants are found in these regions

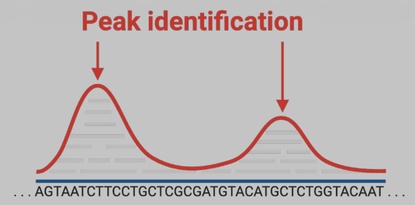
Data

The original paper was trained on the human GRCh37 reference genome using 919 unique transcription factors, which required a huge amount of compute power and time.

Instead, we decided to use **mouse data**, incorporating ChIP-seq data from 39 TFs. We pulled this data from the **ENCODE** project and the **UCSC genome browser.**

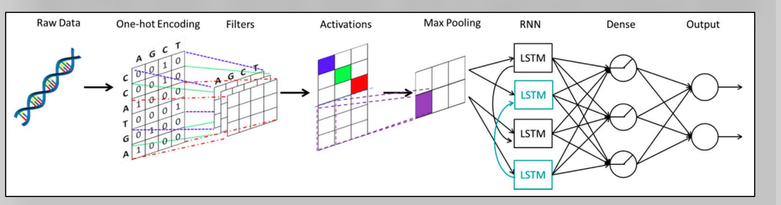
Architecture

Preprocessing: We split the mouse genome into **non-overlapping 200 bp bins** and overlapped them with **ChIP-seq peaks from 39 TFs**. Bins that contained at least one peak were kept and extended 400 bp upstream and downstream to generate reads of 1,000 bp. Labels consisted of 39x1 vectors representing the presence or absence of certain TF peaks in each bin.



[00010000000000...0000100000000000]

Model architecture: The model is a hybrid framework that combined **CNNS** and **BiLSTMS**. The model consists of 5 layers: A convolutional layer, max pool layer, BiLSTM layer, Dense layer, and a sigmoid output layer.

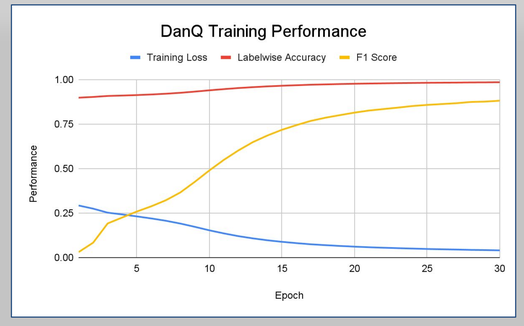


Issues

Preprocessing took a lot more computational power than expected. Our initial attempt totally wiped Phila's computer :(

Our model was able to train successfully, but it seems to have severely overfit the data. We suspect the overfitting arises from excessively complex architecture for our 39 transcription factors compared to the original paper's 919.

Results



- **Training accuracy starts really high** (in the low 90s), probably because of class imbalance – there are way more 0s in our target vectors than there are 1s. As such, **F1 scores are used to provide a more meaningful measure of the model's performance.**
- The model appears to be **sensitive to initialization.** In two separate training runs, F1 scores started out very differently—once as low as 0.003, and another time at a relatively higher 0.14. This was corroborated in the original paper, actually, which was the justification for DanQ-JASPER – a model initialized with known motifs.

DanQ Training Performance	
Average loss	0.041
Labelwise accuracy	0.985
F1 Score	0.881

DanQ Testing Performance	
Average loss	0.397
Labelwise accuracy	0.905
F1 Score	0.301

Model seems to be overfitting; F1 scores during training get really high (~0.85), but are much lower during testing (~0.3).

Future Directions

If we were to continue this project, we'd want to **address overfitting** with model simplification.

Given more time, we also would be interested in doing **model interpretation** to pull out which TF motifs the model is basing its functional predictions on. This kind of meta-analysis would give us greater insight into the biological role of each TF, which could be applied to wet lab research contexts.