Capstone Proposal

Rubén Chaves Moreno

Machine Learning Engineer Nanodegree

January 16th, 2020

Classifying DNA non-coding sequences

Domain background

Genomics, as other life sciences like biology and medicine, is generating big amounts of data. As prices of whole genome sequencing keeps to going down (about $700 for commercial users right now, $100 predicted for 2020), genomics becomes more and more data-driven and it aims to be one of the big contenders of big data[1]. However, to extract the real value, acquiring the data is only the first step, we still have to storage it, distribute it and analyze it.

For this last step, there have been numerous studies since the first full sequencing of the human genome in the Human Genome Project (HGP) where it was estimated that only 1.5% of it, around 20,000 genes, are coding genes. The 98.5% remaining, firstly known as 'junk DNA', as been subject of study in projects like the continuation of the HGP, the Encyclopedia of DNA Elements (ENCODE), where it was found that at least 80% of the DNA is functional through mechanisms as promoters, enhancers, regulatory RNA or chromatin formation regions.

Problem statement

Over 150 genome-wide association studies with 500 trait/disease-associated single-nucleotide polymorphisms (SNPs) located 88% of them in non-coding regions [6], revealing a greater need for predictive algorithm focused on these parts of the genome.

Datasets and inputs

We use the data from the DeepSEA framework (download link), a dataset of the GRCh37 reference genome, segmented in 1,000-bp fragments, with 919 binary targets for TF binding, DNase I sensitivity and histone-mark profiles, from ChIp-seq and Dnase-seq peak sets from the ENCODE and Roadmap Epigenomics data. Each of the four bases (A, G, C and T) are represented with a dimension of one-hot encoded vector. It is divided in training set (4,400,000 samples), validation set (8,000 samples), and testing set (455,024 samples).

As illustrated in [4], it is a highly unbalanced dataset with the majority of target observed in less than 5% of the training samples.

Solution statement

To confront this challenge the use of data driven algorithms, a field known as machine learning, is a perfect fit. While many traditional approaches have been extensively used in bioinformatics, deep learning, a new family of algorithms that have given great results in other fields like computer vision, robotics and many others, is an ideal tool for genomics due to lots of data available, a big and complex problem and no hand crafted features.

More specifically, an especial subset of these algorithms specialized on sequences of words, field known as natural language processing (NLP), give us really powerful tool that easily adapt to our problem. Currently, the best models are based on the original transformer [8] extending it with improvement like the Transformer XL [9], adapted for longer sequences and which we'll use for our genomics data. Additionally, we'll test and specific improvement target for genomics introduced in [7] where they add an extra convolution layer in the attention head.

Evaluation metrics

The dataset has a low number of positive targets, making metrics like accuracy bad for our purpose. We'll focus on:

- ROC AUC: area under the true positive rate (TPR) vs true negative rate(TNR). Bad for our imbalanced data because positive targets are sparse, an algorithm only predicting negative would perform well. We use it to compare with others models.
- PR AUC: precision (PPV) vs recall (TPR). There is no focus on the negative class making it better for imbalanced cases.

$$TPR = \frac{TP}{TP/FN} \qquad TNR = \frac{TN}{TN+FP} \qquad PPV = \frac{TP}{TP+FP}$$

The metrics are calculated for each label and then we aggregate them using a weighted average where weights are the percentages of existence of each corresponding target in the training set.

Benchmark model

The dataset was originally used in DeepSEA[2] and the used in the DanQ[3] and NCNet[4] papers. Metrics are summarized next:

	Accuracy	ROC AUC	PR AUC
DeepSEA	98.21%	0.9046	0.4463
DanQ	98.24%	0.9109	0.4698
NCNet-bRR*	98.35%	0.9441	0.5358
NCNet-RbR*	98.36%	0.9507	0.5519

*Ncnet data is reported with relative values compared to an reimplementation of DanQ(r-DanQ), but they don't give any absolute value from either NCNets nor r-DanQ.

Project design

The main library used will be PyTorch, a deep learning framework that support most of the latest advances giving the user flexibility to make its custom model without sacrificing code readability nor speed and supporting hardware accelerators such as GPUs and half-precision floating-point (FP16). The steps taken will be:

1. Data exploration: looking for the distribution of the target to see how much our data is imbalanced, the total number of positive vs negatives and the distribution of each label, looking if there is any category without positive labels and the percentage of each base.

2. Data preprocessing: the data is already processed and ready to train, the only things to do would be preparing the batching policy and transforming one hot encoding to class labels if we are using embeddings.
3. Model design: code our different models implementations that we'll briefly compare doing short training loops. The main ideas to try are:
    - Hot encoded vs embedding.
    - Convolution 1D, ResNet before the transformer.
    - Implementation of Transformer-XL.
    - Attention head with or w/o intra convolution.
4. Training: using the training and validation test.
    - Compare the different model by data efficiency (loss/epoch) and speed (loss/training time).
    - Hyperparameters tuning: after the try-outs of the models we'll dive into one model and try to search the hyperparameters to speed up the training and maximize our metrics and improve the generalization capabilities of the network.
5. Evaluation on the test set and comparing to the benchmark models.
6. Model interpretation: deep learning models are usually seen as black box algorithms, but we'll look for patterns on the activations of layers and kernels looking for common motifs to get insights of how non-coding sequences work.

References

[1] Big Data: Astronomical or Genomical. Link
[2] Predicting effects of noncoding variants with deep learning–based sequence model. Link Code
[3] DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences Link Code
[4] NCNet: Deep Learning Network Models for Predicting Function of Non-coding DNA. Link
[5] Opportunities and obstacles for deep learning in biology and medicine. Link
[6] Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. Link
[7] Novel transformer networks for improved sequence labeling in genomics. Link
[8] Attention Is All You Need. Link
[9] Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. Link
[10] PyTorch: An Imperative Style, High-Performance Deep Learning Library. Link Web