Classification of Transcription Start Sites in the Human Genome

Ann He, Zandra Ho, Chloe Siebach

1. Introduction

The dramatic advancement in genome sequencing within the last decade has transformed the field of genetics. As our data samples grow exponentially and biological techniques become infeasible, there have been new initiatives to use big data to gain insight into gene regulation. The central dogma of biology states that the flow of genetic information goes from DNA to RNA to protein. Each gene has regions that control the level of transcription, or the level to which DNA is being transcribed to RNA before it is then translated into functional proteins for our bodies. These regulatory regions fall into two major categories: promoters and enhancers. Promoter regions serve as the binding site for the transcriptional machinery (RNA polymerase), while enhancer regions bind molecules that help recruit RNA polymerase to the promoter.

We use data from the Roadmap Epigenomics Project, which used histone modification data to classify transcription start site (TSS) regions into functional categories in several dozen cell types. We hypothesize that the sequence of DNA surrounding each TSS determines a TSS's function, and thus propose to develop a machine learning method that is able to predict a TSS's functional class from its sequence context. We implemented four models (Naive Bayes, Random Forest, Gradient Boosting, and Convolutional Neural Networks) and trained them on TSS and epigenetic data collected from three cell lines (A549, GM12878, and K562). The ability to classify known TSS regions on a large scale would open up many doors for continued exploration of the mechanisms governing gene regulation.

2. Related Work

There is a lot of interest in classifying these TSS regions as promoters or enhancers, but practice has proved difficult. In particular, the ability to distinguish between promoters and enhancers is fundamental in understanding the mechanisms of gene regulation. Epigenetic marks are highly predictive of functional class[CITE chromHMM], but the required experiments are expensive and require large amounts of input material.

In particular, histone ChIP-seq (chromatin immunoprecipitation, followed by DNA sequencing), is one of the leading technologies available for visualizing information encoded in the epigenome. While this method is able to identify the promoters and enhancers well, this technology is too expensive and impractical to utilize on a large scale.

Despite the interest and demand for classification of human TSS's, there have been no attempts to use machine learning for annotations thus far. In addition to expediting the overall annotation process in a cost-effective way, this novel approach would surpass biological assays by generalizing to classify rare cell types as well, an area in which lab techniques have failed due to their dependence on large amounts of biological material. The ability to predict functional class from sequence and experimentally identified TSS's would expand the catalog of known enhancers, and would provide insight into the sequence features that differentiate enhancers from promoters.

3. Data & Features

Our dataset consists of 40,000 TSS regions spread across three cell lines—A549, GM12878, and K562. This data comes from the Roadmap Epigenomics Project, a government-funded effort to provide an epigenomic map to identify key functional elements in the human genome for basic biology and disease research. Having a complete annotation of the promoter and enhancer regions of a cell line will provide crucial epigenomic insights, since these areas interact with transcription factors to upregulate and downregulate protein expression.

It should be noted that although there are many more enhancers than there are promoters in a genome, 92% of our training examples are promoters. This imbalance can be attributed to the method of TSS identification, which relies on the collection of transcribed RNA. These RNA strands arise naturally from promoters, while they only surface due to the accidental transcription of enhancers. The statistical power to detect promoter TSS regions is reflected in the data set.

3.1 Preprocessing

3.1.1 Region size

A TSS region is determined using information gathered from numerous assays of a cell line. These assays pinpoint the exact base pair that is transcribed first, and this exact location has been found to vary. The slight variation leads us to define a general region—the transcription start site region—that on average spans approximately 40 base pairs. We relied on Dr. Nathan Boley's peak calling pipeline to identify these regions (Boley). The DNA responsible for attracting various proteins for transcription surrounds the TSS region. To train our classifiers, we considered region sizes of 500 and 1000 base pairs with the TSS at its center.

3.1.2 K-mers

K-mers were used to preprocess the data for Gradient Boosting and Random Forest. In computational genetics, the k-mers of a DNA sequence refer to all its possible subsequences of length k. These k-mers therefore keep track of consecutive patterns of length k in the data, and though they do not track larger or non-contiguous patterns, they are often used for convenience and for where continuous patterns are enough to classify accurately. For each TSS sequence in our data set, we created a feature vector containing the counts of distinct 6-mer appearances of all consecutive 6-mers of that sequence. We chose to use 6-mers since 4^6 = 4096 distinct permutations gave us a large spectrum of distinguishing patterns to classify along, without causing our dataset to be too large to work with. The observation of a specific sequence of k = 6 base pairs is long enough to be meaningful (as opposed to something as short as k=1, which would just return a count of each base in the sequence), and yet short enough to appear frequently. The k-mer vector is analogous to the vector of word counts used for classifying spam and non-spam emails.

3.1.3 One-hot Encoding and Filters

One-hot encoded sequences are used to pre-process the data for Neural Nets. Because there are four options for which a base is at any location, this process takes a sequence of length n and converts it into a matrix of size 4 x n, with each row corresponding to a base (A, C, G, or T). The values of the matrix M are indicator variables, such that M(A,j) = 1 if the jth base is A, and 0 otherwise. An example encoding is pictured in figure 1.

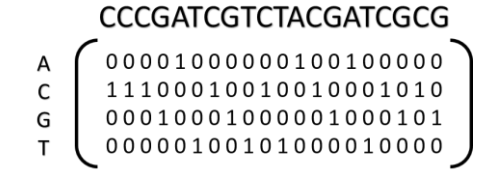

Figure 1. An example one-hot encoding sequence, to be fed into a Neural Nets model.

The first layer in our neural nets architecture is a convolutional layer, which scans the one-hot encoded sequences with filters of size 4 x 32. Over the course of training, these filters identify patterns in the encoded sequences that are important for predicting labels. Neural nets also iterates through several more "hidden layers" of data processing, but the first is the most useful to understand because the trained filters can give us insights into base-pair patterns that differentiate between enhancers and promoters.

3.1.4 Balanced Data Sets

As mentioned previously, our data set contains almost 10 times as many promoters as enhancers. We accounted for this disparity by introducing balanced data sets while training our classifiers, drawing a subset of promoters at random to create a new train set.

4. Methodology

4.1 Model Selection

Initially, we considered Naive Bayes, Random Forest, Gradient Boost, and Convolutional Neural Networks as models. We quickly found, however, that the Naive Bayes' assumption of independence between k-mers made for poor predictive ability, and chose to focus only on the other three. Within each of those models, we examined the impact of model parameter choice on accuracy using cross validation techniques on balanced data sets, and then trained each model using the optimal parameters on the entire data set.

4.1.1 Gradient Boosting

For gradient boosting, we used the logistic regression loss function and primarily examined the impact of changing the number of weak learners that it considered in its predictions. Again using 10-fold cross-validation, we found that the best accuracy occurred with 300 weak learners (figure 2). Here, accuracy was measured by area under the ROC curve.

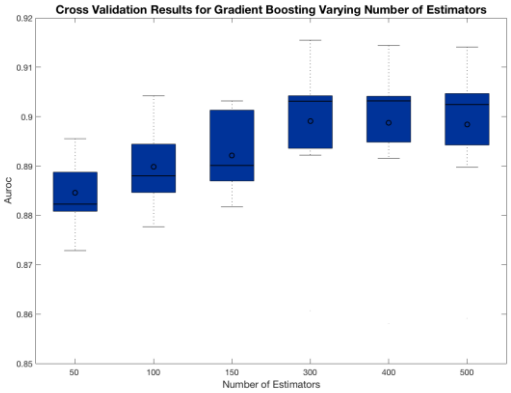

Figure 2. Cross validation results for Gradient Boosting with varying numbers of estimators, with accuracy measured by AUROC.

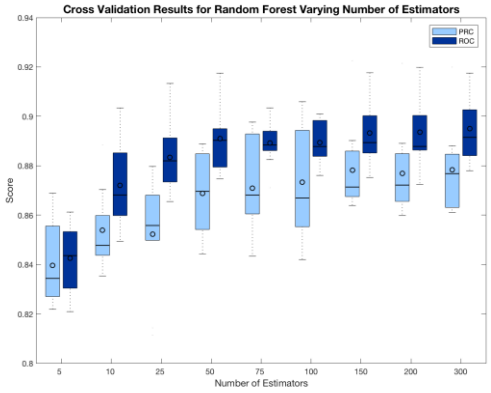

Figure 3. Cross validation results for Random Forest with varying numbers of estimators, with accuracy measured by both AUROC and AUPRC.

4.1.2 Random Forest

Random forest models are parametrized by the number of trees that are built during training and the depth of those trees (among other parameters). We quickly found that changing the depth of the tree did not have a significant impact on our model, and chose instead to focus on the number of trees. Using 10-fold cross-validation, we found that the best accuracy occurred when we used 75 predictors (figure 3) with the logistic regression loss function. We began by evaluating model performance based on the area under the receiver operating characteristic curve. In anticipation of an unbalanced test set, we later considered the area under the precision-recall curve as well.

4.1.3 Convolutional Neural Networks

Convolutional Neural networks have been shown to work well on DNA data in the past, so we quickly found architectures and parameters that performed decently well (~0.70 AUROC, ~.90 AUPRC) on cross-validation test data. The two architectures can be found in figure 4, and the plot of their relative predictive accuracies can be seen in figure 5.

Architecture 1:
Convolution
Dropout
Flatten
Dense
Dropout
Activation
Dense
Dropout
Activation
Dense
Activation.

Architecture 2:
Convolution
Dropout
Maxpool
Convolution
Dropout
Maxpool
Flatten
Dense

Figure 4. The sequences of layers used in the two different neural nets architectures that were considered.

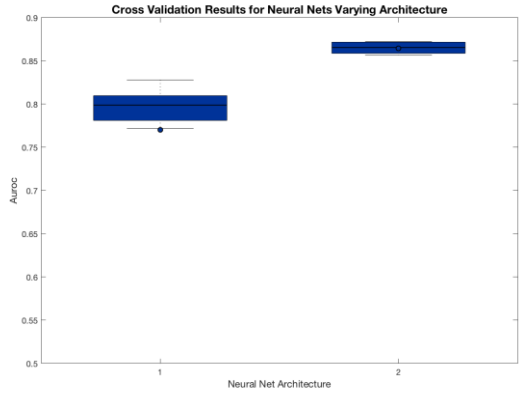

Figure 5. Comparison of cross validation results for different neural nets architectures, with accuracy measured by AUROC.

As we trained the two differently-architectured models, we used a callback function of Keras neural nets software that allowed us to stop training the model when the performance on a hold-out data set began to worsen. This method ensures that the model learns as much as possible from the training data without overfitting. To understand this method more

fully, we made a plot of loss (training and validation) vs epoch (training iteration) for both models (figure 6). The loss function used was binary crossentropy. The plot shows that the training loss generally decreases, whereas the validation-loss jumps around and eventually starts to increase. Once this increase starts to occur, however, the Keras package stops training the model, preventing overfitting.

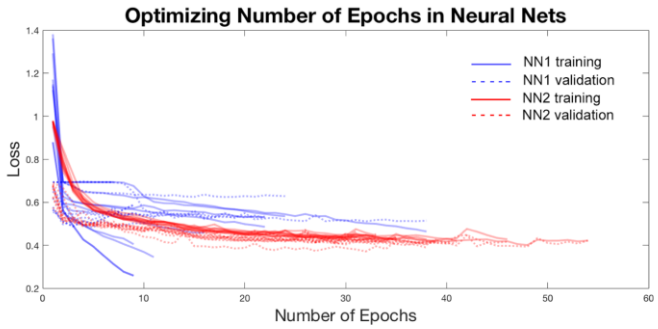

Figure 6. Change in the values of training and validation loss for neural nets architectures as a function of the number of training iterations run.

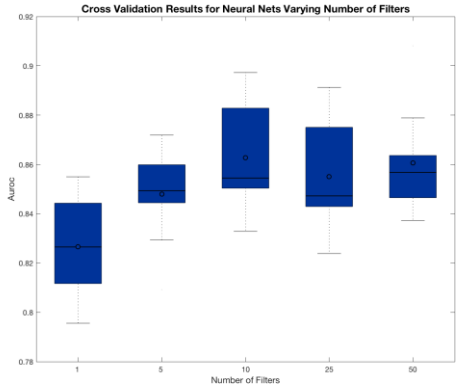

Figure 7. Cross validation results for Neural Nets models with varying numbers of filters, with accuracy measured by AUROC.

Based on the information gathered, we chose to further tune and analyze model architecture 2. The boxplot in figure 7 shows the impact of varying the number of filters in the first convolutional layer on the accuracy of the model. We wanted to have as few filters as possible while still optimizing for accuracy (measured by AUROC). After experimenting, we chose to proceed with the model that used 25 filters. We were ultimately able to fine-tune our model with great accuracy (~0.98 for both AUROC and AUPRC on cross-validated test data) thanks to the experience of our mentor with this kind of data.

5. Results

After training our final Gradient Boosting, Random Forest, and Convolutional Neural Networks models on the A549, GM12878, and K562 cell lines, we tested on the HEPG2 cell line.

5.1 Gradient Boosting and Random Forest

We discuss the results from the Random Forest and Gradient Boosting models together due to their similarity in weighting the importance of all 4096 possible 6-mers. Figure 8 displays the top 10 6-mers identified by each model. Although we cannot conclude from the lists that a particular sequence pattern can be used to identify promoters, we do learn that the guanine and cytosine nucleotides are highly indicative of promoter regions. This result is consistent with the literature, as sequences of guanine and cytosine tend to form strong intermolecular forces that would promote the binding of RNA polymerase to the region (Kudla et al). The top-10 lists generated by the two classifiers share four kmers in common, which was an encouraging sanity check. Both models did very well on area under the precision-recall curve (GB: 0.988, RF: 0.989) and less well on area under the receiving operator characteristic curve (GB: 0.881, RF: 0.885).

Random Forest		Gradient Boosting	
GCGGCG	GGGCGG	CGCCGC	GGCGGC
CGCCGC	CCGCCC	CCGCCC	GGGCCC
CCGCGC	CGGCCG	GGGCGG	GCGCCG
CGCGGC	CGCGCG	CGGCGG	GGGGCC
CCGCCG	CCGCGG	GCGGCG	CAGGCA

Figure 8. Top ten 6-mers for the Random Forest and Gradient Boosting models. 6-mers found in both models are highlighted in blue.

5.2 Convolutional Neural Networks

We chose to interpret the filters produced in the first layer of the architecture as it deals directly with the data, and is trained to detect patterns that appear very frequently or infrequently in promoter regions. Our 25 filters are initialized (mostly) randomly and orthogonally to capture as much variety as possible, and the training process changes them to be representative of significant patterns in the data. Admittedly, many of the filters were noisy and difficult to analyze, and we chose to display the two most representative filters in figure 9. Consistent with findings from the Gradient Boosting and Random Forest models, our Neural Nets classifier recognized cytosine and guanine nucleotides in close proximity. This model also revealed an important motif consisting of adenine and thymine content. This finding is also consistent with the literature, as high AT content corresponds to flat regions of DNA--ideal sites for the unwinding of the double helix for transcription (Rohs et al). The neural networks model turned out to be highly accurate, doing extremely well on both AUROC (0.947) and AUPRC (0.994).

Convolutional Neural Networks classifier outperformed the Gradient Boosting and Random Forest classifiers, suggesting that Neural Nets probably recognized some features that were not present in the k-mer lists. This result exposes an interesting limitation of the contiguous k-mers used in Gradient Boosting and Random Forest. Consider motif #2. The nucleotides are not clustered together, but rather span a region of 20+ base pairs, which Gradient Boosting and Random Forest would understandably fail to recognize.

Random Forest, Gradient Boosting, and Convolutional Neural Networks all have different underlying mechanisms of learning from a data set, but they each reveal the importance of GC clusters for identifying promoter regions. This result is consistent with what could have been expected, because GC content is related to how strongly enzymes are able to bind to that region of DNA. Additionally, our Convolutional Neural Networks model suggests that intermittently-spaced adenine and thymine nucleotides are also highly predictive of promoters in transcription start sites--a feature that is unrecognizable by the 6-mers of Gradient Boosting and Random Forest. All three models performed very well on the test set, confirming our initial hypothesis that models trained on certain cell lines may be generalized across many cell lines. Our results indicate that machine learning is a good, cost-effective alternative to current biological techniques of transcription start site classification.

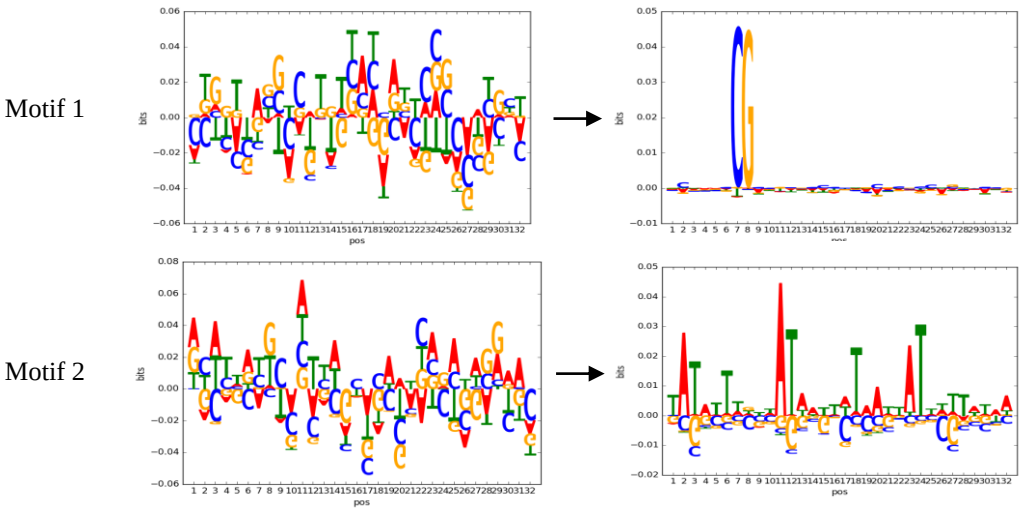

Figure 9. Motifs of two of the filters used on the final neural nets model, before training and after training.

References

Boley, Nathan. (2014). Methods for the Analysis of High Throughput Sequencing Data. UC Berkeley: Biostatistics. Retrieved from: http://escholarship.org/uc/item/51b862fc

Kudla, G., Lipinski, L., Caffin, F., Helwak, A., & Zylicz, M. (2006). High Guanine and Cytosine Content Increases mRNA Levels in Mammalian Cells. PLoS Biology, 4(6), e180. http://doi.org/10.1371/journal.pbio.0040180

Rohs R, West SM, Sosinsky A, Liu P, Mann RS, Honig B. The role of DNA shape in protein-DNA recognition. *Nature*. 2009;461(7268):1248-1253. doi:10.1038/nature08473.