Deep Learning for Natural Language Sequence Labelling Applied to Epigenomics

Seth Hildick-Smith
Department of Computer Science
Stanford University
Stanford, CA
sethjhs@cs.stanford.edu

Ivaylo Bahtchevanov
Department of Computer Science
Stanford University
Stanford, CA
ivaylogb@cs.stanford.edu

Abstract

Foundational to our understanding of DNA meaning is our ability to model its function. With the aim of creating a predictive model on epigenome DNaseI-accessible enhancer regions we apply NLP techniques to the hereditary language. With the tools borrowed from the NLP domain, we make strides towards an accurate model for the location and cell type associated with enhancers in our sequence dataset. Experimentation with data embedding and model architecture lead us to narrow the range of successful models for this task. The most successful model we studied achieves a balanced accuracy of 73% and a gain in certainty of 1.46 on our test data. We know of no previous attempt to build a predictive model on this dataset.

1 Introduction

Although cells of eukaryotic organisms share the same fundamental DNA, these cells may express vastly different phenotypes. The wide diversity of cell structure within these organisms speaks to the complexity of gene expression. We pursue the study of epigenomics, that of the complete set of hereditary material in chromatin and its expression, through the lens of Natural Language Processing. Deep learning for NLP provides us with the framework to process the language of the chemical processes by which human cells define their state.

Our primary task comes down to sequence labelling, a pattern recognition objective that entails the algorithmic assignment of a categorical label to each member of a sequence of observed values. In our case, this labelling refers to assigning positive and negative values to segments of a DNA sequence that determine active vs. nonactive (respectively) DNaseI enhancer regions. We perform this classification task for each of 57 "high quality" cell types.

We apply four deep learning architectures which have demonstrated success in related biocomputational tasks to the epigenomics task at hand. These deep architectures are trained on data sourced from Stanford Medical School's Kundaje Lab to predict regions of DNA shown to be DNaseI-accessible enhancer regions. We validate and test the distribution learned in our models through a 70-20-10 training, validation, testing segmentation of the dataset.

2 Related Work

In our literature review we examine the most successful deep learning architectures in the field of genomics in the hopes that applying similar models to our task will yield promising results.

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

In the DanQ paper [4], the authors build a "hybrid convolutional and bi-directional long short-term memory recurrent neural network" to better predict non-coding DNA function. This model builds on the paradigm presented by Yandell and Majoro [3] by using models typical of of Natural Language Processing. Under that perspective, they set out to predict function of DNA with a CNN/RNN mixed architecture. The authors work in a deep learning paradigm - taking input directly from sequence.

Each input vector to the DanQ model is a 1000 base pair series centered on a single data parse (200 base-pair sequence). Each input vector that overlaps more than one TF binding ChIP-seq peak is paired with the respective target vector for training. Note that about 10% of all target vectors were all negatives. The authors further parsed all input into a binary representation by adding 4 additional columns for each nucleotide one each corresponding to A, G, C and T.

The authors focus nearly exclusively on comparison against the DeepSea [6] model as that the two models shared input data. The authors found that their model outperformed DeepSea on 94.1% of the input-target pairs. It is valuable to note that the difference between the two models is comparatively small with an absolute improvement of around 1-4% for most targets. When the authors compared their work to DeepSea under a precision-recall curve analysis, DanQ outperformed the other model by over 50%.

2.2 DeepSea: Predicting effects of non-coding variants with deep learning–based sequence model

Building off the research done by Quang and Xie [4], Zhou and Troyanskaya [6] work to advance the understanding of non-coding DNA. That is, given coding DNA, the authors work to produce a model that is effective enough at predicting the function of that DNA sequence to extract the function of non-coding DNA de novo. Given such a model, the authors would have a tool to aid in understanding the 98% of human DNA that is non-coding.

The authors here note that the 200 base pairs which required at least one TF binding event, represented in total, 521,636,200 base pairs of sequences. This is about 17% of whole genome. The authors then split this data into training and evaluation sets for chromatin feature prediction performance.
The authors took the high dimensional input tensors into their model to predict the 919 classes in their multi-class classification.

Their model is a pure convolutional network sequence built to capture varying length motifs in the genomic sequences. The model, more explicitly uses three convolution layers followed by a fully connected layer followed by a ReLu and a sigmoid output over the probabilities. The authors clarified the 919 output dimensionality as all chromatin features: 125 DNase features, 690 TF features, and 104 histone features.

The DeepSea model is primarily compared with a previously implemented gapped kmer SVM model that attempted to achieve an equivalent goal. DeepSea outperformed gapped k-mer SVM (gkm-SVM) on transcription factor binding prediction in all cases. The model also achieved a higher area under the precision-recall curve in almost all cases. Furthermore, the linear model did not gain performance from increasing size of context sequences to the size equivalent to that of the deep learning model.

2.3 Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks

The "Basset" model proposed by Kelley et. al. [2], offers a powerful computational approach to annotate and interpret the non-coding genome. The authors argue that just considering the overlap of a variant with annotations will under-utilize the sequence data, and thus produce an inefficient model. They defend that understanding the DNA-protein interactions can be done as a function of the underlying sequence, thus creating additional features not immediately understood by a simple deep learning model. The signals can be interpreted through NLP tasks that learn what effectively translates to sentence syntax in normal language in the form of genetics.

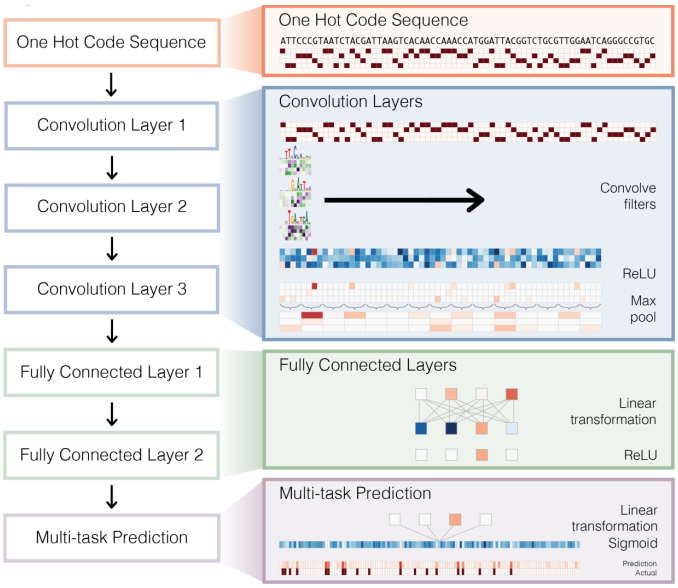

Figure 1: Schematic overview of the Basset framework, a model built to predict the chromatin effects of non-coding DNA.

They use the training data to build a model to effectively predict protein binding, DNA accessibility, and DNA methylation from the sequence. The trained model then annotates the influence of every nucleotide (and variant) on these regulatory attributes. The authors apply convolutional neural networks (Fig. 1) to learn functional activities of DNA sequences.

These CNNs have proven highly effective in a number of diverse tasks; this set recently includes biological sequence analysis. Rather than choose features manually or in a pre-processing step, convolutional neural networks adaptively learn them from the data during training. They apply nonlinear transformations to map input data to informative high-dimensional representations that trivialize classification or regression. In a convolution layer, the algorithm scans a set of weight matrices called filters across the input; these weight matrices learn to recognize relevant patterns.

For DNA sequences, the initial convolution layer corresponds to optimizing the weights of a set of position weight matrices (PWMs), which are a well-studied tool in bioinformatics. These PWM filters search for their motifs along the sequence and output a matrix with a row for every filter and column for every position in the sequence.

The Basset model used two key functions for optimization: computing the loss and gain scores for every nucleotide. Loss score is computed as the predicted activity with referenced nucleotide minus the minimum predicted activity after permuting positions with the other 3 nucleotides.

On the other hand, the gain score is the max predicted activity after the permutations minus the referenced nucleotide activity. Their training accuracy is 80 percent, but their test falls down to 46 percent, indicating that their training regime may have entered an over-fit space, and is able memorize patterns through feature extraction as opposed to learning syntax and structure of the sequences.

3 Approach and Models

3.1 Basset

We apply the "Basset" model proposed by Kelley et. al. [2]. The model consists of three convolutional layers (300 kernels, window size 19; 200 kernels window size 11; 200 kernels, window size 7) followed by a 2-deep multilayer perception (each layer with 1000 neurons) and a dense output layer. Each of the three convolutional layers apply a rectified linear activation and batch normalization to

aid deep training. Between convolutional layers the model applies max-pooling to regularize and reduce data depth.

3.2 Greenside-Basset Variation

Although our literature review indicates the promise of the Basset model, guidance from the Kundaje lab indicated that some of their tweaks to the model achieved better results. We applied the Greenside-Basset Variation model developed in that lab to our data. This model applies more mid-level kernels in the convolutional layers and an additional fully-connected layer at the end of the model. Specifically, the three convolutional layers are defined with 250 kernels, window size 4; 500 kernels window size 4; 250 kernels, window size 7. Each of the subsequent layer of the MLP consists of 1000 neurons each.

3.3 DeepSea Variation

We applied a variation of the DeepSea model. The original model was initially not suited to the task due to a risk of overfitting, so we experimented to find ways to tailor it. To that end, we removed a convolutional layer from the bottom of the structure and reduced the number of filters at each layer. The model we ultimately tuned had three convolutional layers (256 kernel, window size 16; 128 kernel, window size 16; 64 kernel, window size 16) followed by a 2-deep multilayer perception (each layer with 1000 neurons). This variation was the best version of the DeepSea experiments, since its simplicity opened itself for a more general task, whereas the specific model did not lend itself to sequence labelling the cell type activations.

3.4 DanQ Variation

The DanQ model presents a hybrid model that begins with a conv-batchnorm-relu-maxpool segment, followed by a sequence of LSTM layers, and finally a dense layer. The model's convolutions take a bird's-eye view of the sequence while the recurrent layers learn inherent representations with the surrounding base pairs. The initial model was made for more general genomics tasks, but we were able to fine-tune it to the sequence labelling task. This variation is thus more suitable for identifying the representations signifying active regions.

At a technical level, the DanQ model is comprised of a single convolution layer (320 kernels. Window size: 26. Step size: 1.) followed by a pooling layer (Window size: 13. Step size: 13.), then a bi-directional long short term memory layer (320 forward and 320 backward LSTM neurons) a fully connected layer (925 neurons) and a sigmoid activation. Note that the original authors trained two sizes of the DanQ model, we studied the smaller of the two.

4 Experimentation and Results

4.1 Dataset

Our dataset is a portion of the Delineation of DNaseI-accessible regulatory regions dataset made available by the NIH Roadmap Epigenomics Mapping Consortium. We gained our full data set through the generosity of the Kundaje Lab.

The data consists of the human GRCh37 reference genome represented as a 3Gb text file of nucleobases and two sets of .bed files which contain the active regions for 128 cell types each. To reduce the quantity of data we dropped the 128 cell types to 57 "high quality" cell types indicated to us by the Kundaje Lab. The two label sets of .bed files correspond to high recall and high precision labels. We segmented our label region into a series of mutually exclusive ranges of 200 nucleotide sequences. To provide context for our model we expanded each 200 base pair sequence on either side by 400bp. Thus, the target labels were length 57 binary vectors, one such vector for each 200 nucleotide centered input sequence. The inputs were embedded 1000bp (C,A,G,T) sequences.

After reducing our data to high quality inputs, we have 2,713,213 size (1000$\times$embedding depth) samples each with a corresponding 57 dimensional label vector. Note that the label vector is not one hot: a DnaseI enhancer region may be active in more than one cell type.

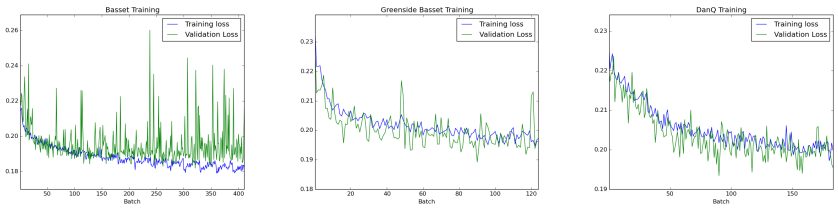

Figure 2: The most significant training/validation loss regimes we explored. Note the Basset model remains on the verge of entering an over-fit space throughout training while none of the other models suffer from that risk.

4.2 Embeddings

We have experimented with two methods of embedding our input. The first is a one-hot encoding of the genomic characters. The second is a 2-dimensional embedding that captures mutually bound nucleobases along the same axis. In this compressed representation, one axis is represented by a linear 1 or -1 for A or G respectively, the other the same for T or C respectively. The first option has the benefit of removing all linear relations among the different elements of the sequence, while the second uses half as many parameters. Since there may be non-obvious, non-linear relations between the mutually bound nucleobases, we chose to commit our constrained training time to the deeper, better studied embedding as with that embedding a model has the ability to learn the relationship encoded in the other.

4.3 Loss Function

Initially, we began our modeling applying a binary cross-entropy loss that penalized nodes based on knowledge from only the high precision label set. As our research continued, we built a second set of labels derived from the high recall label set. To merge the two, we kept the high precision labels except when a given enhancer is found in the high precision set but not in the high recall set. These exceptions are denoted "ambiguous" with the value -1. Thus, for each candidate input sequence we have a label vector $y \in \{-1, 0, 1\}^{57}$. To capitalize on the captured, ambiguity we apply a modified binary cross entropy loss: Let σ denote the sigmoid function and θ denote the output of the last layer of the architecture.

$$\hat{y} = \sigma(\theta) \tag{1}$$

$$BCE'(y, \hat{y})_i = \begin{cases} -\big(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\big) & y_i \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We apply the four models, each a modification to either the Basset, DeepSea, or DanQ architectures, to minimize the modified cross entropy loss function.

We implement our models on a Theano [5] backed Keras [1] environment. We support our experimentation with eight NVIDIA GRID GPUs on two AWS instances.

4.4 Evaluation

Our label set is fundamentally very skewed: we found that a simple majority class predictor achieves 93.8% accuracy. To address this natural bias towards high accuracy we instead choose to evaluate on balanced accuracy, a metric for which the naive majority class predictor achieves 50%.

$$\text{balanced accuracy} = \frac{\text{sensitivity} + \text{specificity}}{2} \tag{3}$$

$$= \frac{0.5 \times \text{true positives}}{\text{true positives} + \text{false negatives}} + \frac{0.5 \times \text{true negatives}}{\text{true negatives} + \text{false positives}} \tag{4}$$

Additionally, we examine the sum of sensitivity and specificity: the gain in certainty which ranges from 0 to 2. Any value over result of 1 indicates that the diagnostic test does adds value above guessing.

The value of our epigenetic labelling comes from the ability to deduce which DNaseI enhancer regions are active for each cell type, so we built our models to focus on evaluating the success of those predictors. We evaluate over a threshold and report the set of metrics that maximized the balanced accuracy. We show a detailed comparison of the results for all models across all metrics. See Table 5 for results (Note - the majority predictor is only included as a point of reference).

4.5 Basset

This model achieved the best balanced accuracy of 72.8% along with the highest precision and f1 scores. The model still weighted the negative examples very heavily, as demonstrated by the fact that precision is not particularly strong. The model made an attempt to balance the false negatives and false positives, so its recall, as a result, was slightly lower than the Greenside variation and the DeepSea.

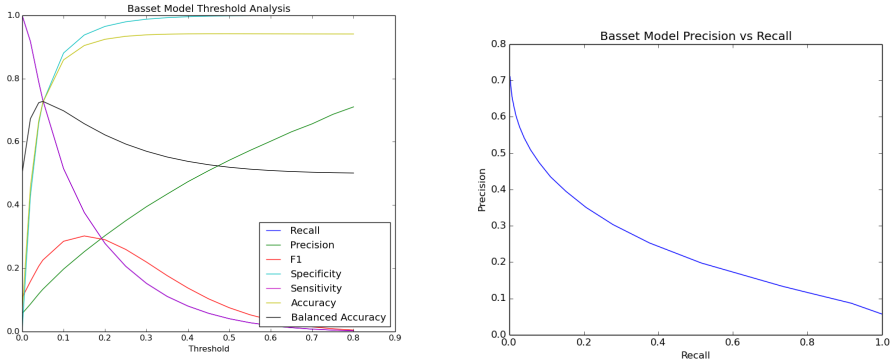

Figure 3: Trained Basset model performance on test data over a range of acceptance thresholds.

4.6 Greenside-Basset Variation

Although this model fundamentally has more expressive power than the original Basset structure, we were unable to successfully use that expressively to our advantage. We achieved a balanced accuracy of 69.99% with this model. We believe the failure to out-preform the Basset model comes down to a more significant training time requirement needed to fit the model and its expanded number of parameters.

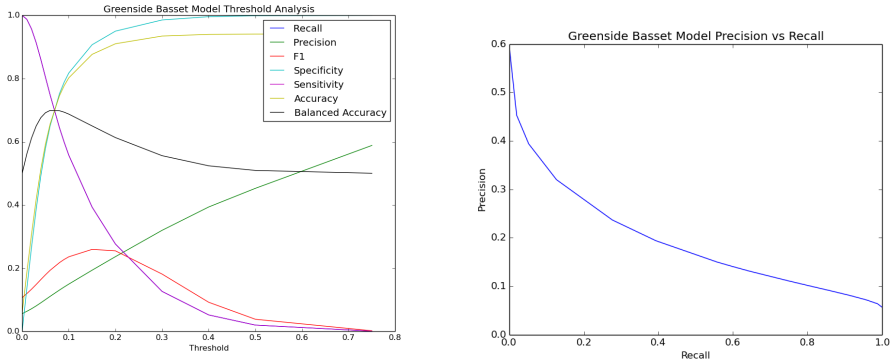

Figure 4: Trained Greenside Basset model performance on test data over a range of acceptance thresholds.

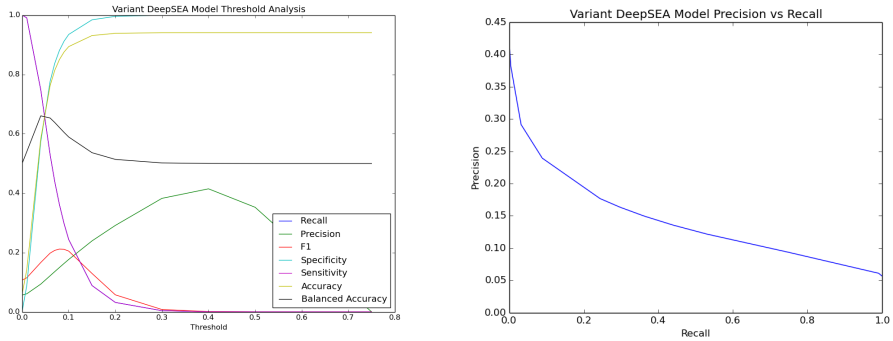

Figure 5: Trained DeepSea model performance on test data over a range of acceptance thresholds.

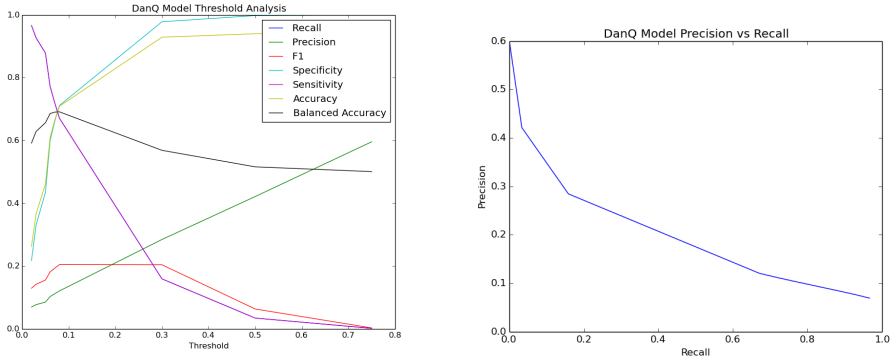

Figure 6: Trained DanQ model performance on test data over a range of acceptance thresholds.

4.7 DeepSea Variation

This model achieved the worst results out of the four we examined. We believe this is in a large part due to the heavy pruning we applied to the network. The original model was constructed to learn a regulatory sequence code from large-scale chromatin-profiling data, enabling prediction of chromatin effects of sequence alterations with single-nucleotide sensitivity. This parallel task translated well to an even-balanced data set, but as demonstrated by the non-existent precision, the model did not successfully overcome the class imbalance.

4.8 DanQ Variation

The DanQ model was the most unique of the architectures, fully leveraging the convolutions and the bi-directional LSTMs. The model, despite having a more complex structure than the others, performed very similarly to the better of the Basset variations. As shown in figure 6, the metrics follow a very similar trend as the Basset.

4.9 Training Optimizations

The dataset we used is large, representing over 12% of the human genome. Input data is large enough that we could not fit the entire X matrix in main memory. To train any of our models, we developed a batching scheme to load anywhere from 17% to 30% of the data into memory at a time. Thus, one epoch of training took up to 60 large disk IO operations. Some architectures (especially our DeepSea variation) were particularly memory intensive. In these cases, we further alleviated memory pressure by taking smaller batches of data from main memory in our gradient update.

	Majority Predictor	Basset	Greenside-Basset	Var. DeepSEA	DanQ
Accuracy	0.9378	0.7244	0.6566	0.5834	0.7088
Precision	N/A	0.1334	0.1112	0.0936	0.1204
Recall	0.0	0.7314	0.7479	0.7481	0.6708
Specificity	1.0	0.7240	0.6508	0.5730	0.7112
Bal. Acc.	0.5	**0.7277**	0.6994	0.6605	0.6910
F1 Score	N/A	0.2257	0.1936	0.1663	0.2042

Table 1: Statistics for each model polled at the argmax for balanced accuracy over threshold. Observe that the Basset model achieves the highest balanced accuracy as F1 score while all model achieve gains in certainty above 1.

5 Conclusion

These signals can be interpreted through NLP tasks that learn what effectively translates to sentence syntax in normal language in the form of genetics. This is why LSTMs can perform reasonably well as encoders for the sequence to learn features similar to standard language. The models we created produced good results for predicting the activations, which shows promise that deep learning can be fitted for the task. Since this was the first exploration into this specific task and data-set, we are very optimistic for future results in the field.

The next steps will tackle a much more tailored dataset to fine tune the precision while maintaining (if not improving) the same balanced accuracy and recall metrics. In reality, the genome will show very sparse activations, and it is crucial that any models tackling the task are equipped specifically for the heavy skew.

6 Acknowledgements

We would like to gratefully acknowledge Avanti Shrikumar and the Kundaje Lab for generously sharing their time and data. Without Avanti's remarkably patient guidance, this project would not have been possible. Additionally we want to thank Peyton Greenside for the Basset alternative architecture, as well as Francois Challet for Keras and his dedication to the platform.

References

[1] François Chollet. *Keras*. https://github.com/fchollet/keras. 2015.

[2] David R Kelley, Jasper Snoek, and John Rinn. "Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks." In: (2015). DOI: 10.1101/028399.

[3] Mark D. Yandell William H. Majoros. "Genomics and Natural Language Processing". In: (2002). nature reviews: 1506.01497 (cs).

[4] Daniel Quan and Xiaohui Xie. "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences". In: *Oxford Journals Nucl. Acids Res.* (2016). DOI: 10.1101/032821.

[5] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: http://arxiv.org/abs/1605.02688.

[6] Jian Zhou and Olga G Troyanskaya. "Predicting effects of noncoding variants with deep learning–based sequence model". In: *Nature Methods* 12 (2015), pp. 931–934. DOI: 10.1038/nmeth.3547.