CS230

Don't Drop the Base Pairs

Brandon Benson
Department of Applied Physics
Stanford University
bensonb@stanford.edu

Alex McKeehan
Department of Physics
Stanford University
mckeehan@stanford.edu

Abstract

In tackling genetic blood diseases such as Leukemia, understanding the genetic mechanisms responsible for abnormal cell expression is a crucial step towards finding a cure. Currently these genetic mechanisms are incompletely understood due to the complexity of genotypic expression and manifestation as phenotypes. Indeed, there are an intractable number of unique genotypes and at least 18 different cell types that comprise the blood. In this work, we aim to expand understanding of this complexity by using deep learning to learn a model that maps between genotype and phenotype. Specifically, we find a map between short gene sequences of 1000 base pairs to a close intermediate metric of phenotype called *chromatin accessibility* (CA). Gene sequences are one-hot encoded as narrow images of width 4 for each of the possible base pairs (A, C, T, G) and a convolutional neural network (CNN) is used to output an array of 18 binaries that approximate CA for each of the 18 cell types found in the blood. We achieve a reasonable CA prediction accuracy of 0.84 and an auPRC of 0.49. We take additional measures to interpret the significance of the trained gene mapping through additional methods including confusion matrices, sensitivity-specificity curves, and a Fourier decomposition by base pair length.

1 Introduction

Recent advances in the theoretical understanding of deep learning and the efficiency of training neural networks have coincided with similar breakthroughs in the biological sciences, enabling low-cost sequencing of the full human genome at base pair resolution. The ability of neural networks to make predictions based on large volumes of data has great promise in areas of genetics, particularly in yielding a better understanding of what genotypes are associated with respective phenotypes. For example, by training a neural network to associate specific gene regions with inherited proclivity to disease or other genetic factors, it may be possible to treat those areas of the genome with gene therapy.

The goal of this project is to determine the gene patterns in the human genome that are most strongly associated with multiple myeloma. We obtained as input a full human genome, hg19, sequenced at the base pair level. The output consists of a binary (0 or 1) classification of the chromatin accessibility (CA) of each part of the genome in segments of length 1,000 base pairs for 18 different cell types. The 18 cell types (Figure 3) are key clinical indicators in the development of leukemia, as an imbalance in the relative numbers of these cell types is a sign of flawed blood cell differentiation. By training a model to predict the CA of each cell type, we hope to provide a means of predicting patient susceptibility of an individual to leukemia based upon their genetic expression in the relevant variants. Similar work was completed by Kelly et. al [15], who developed an efficient tool for making genetic models named "Basset" but was not specifically applied to detecting those genetic sequences of blood

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

diseases. Finally, we explore our resulting CNN model through various methods in the Results and Discussion section.

2 Related work

Many previous efforts have sought to determine genetic mechanisms and emergence of cell phenotypes. Beginning in 2004, Beer et al. [13] applied complex rules using AND, OR, and NOT logic along with intelligent constraints. This systematic genome-wide approach was successful in learning the underlying gene expression and yielded interpretable results. However, it lacks the efficiency and expressive power of deep learning, and so it is overshadowed by results such as [12] or [15]. Alipanahi et al. [12] made leaps of progress in finding genomic sequence specificities of DNA-binding proteins. Similar to our work, Alipanahi et al. uses a convolutional network to map genetic sequences to an intermediate metric for cell phenotype. The work uses a smaller network, however, consisting of a single convolutional layer, a rectifying layer, a pooling layer, and an output fully-connected layer to achieve 0.93 test AUC. Analysis of the network focused mainly on short genetic sequences known as motifs that maximized network activation. These motifs were compared with known genetic sequences of importance and many matches were discovered; we take a similar approach to test the significance of found sequences in our work. Finally, Kelley et al. [15] created an open-source platform for motif identification by using a deep CNN architecture. Most notably, their open source machine learning model, "Basset," can determine both important motifs and single base pair mutations called SNPs, providing the foundation for our work. Specifically, Kelley et al. employed a CNN that could be understood using genetic motifs and base pair sensitivity.

It is these CNN architectures ([12],[15]) that we expand upon in this work. Following a similar approach, we fix the weights and train on the input to determine the base pair sequences that maximize network activation, in a fashion reminiscent of that introduced in "Basset." Next, we explore the trained model parameters in a novel way using confusion matrices to begin understanding the hematic (blood related) cell structure. Previously, CNNs have been shown to learn class hierarchy [19]. Bilal et al. use a large confusion matrix to demonstrate that images of animals that are closer in class hierarchy are more difficult for the CNN to distinguish, particularly over a wide range of animal pictures. Here, we complete an analogous case study of the phenomena, and then continue to other analysis techniques such as training on the input described in the class slides [14]. Visualization of deep neural networks using this technique has previously been studied extensively in works such as Mahendran et al. [16].

Lastly, we attempt to identify key genetic sequences in our model, 'motifs,' using TomTom software [20]. TomTom is a comprehensive tool used for nucleic acid research that matches input genetic motifs to known motifs. "Basset" work also leverages this tool using the "human CIS-BP" database to match motifs, and we use the same database [15].

3 Dataset and Features

The input values for our dataset consist of a full base pair-resolution sequencing of an individual treated for Leukemia, with binary CA for each contiguous sequence of 1,000 base pairs. While we had access to the entire sequence of all 23 chromosomal base pairs, we chose to train solely on chromosome 1, which consists itself of 767,928 input examples. Out of those examples, we chose a set of 10,000 randomly shuffled input examples to use for training our neural network, and split these 70/20/10 into training, dev, and test sets.

Our baseline input data set was provided by Peyton Greenside through the Kundaje Lab at Stanford[7], who provided us with the encoded base pair sequence of the entire human chromosome with associated chromatin accessibility binaries. Preprocessing the dataset consisted of converting the input BED files associated with chromatin accessibility values into characters (A, C, T, G) representing the respective base pair sequence of the genome. The inintial data was provided in the BED format, a file type used to encode a genomic pattern. After downloading the full human genome reference sequence[10], we used bedtools[11], a suite designed for genomic arithmetic, to convert the genomic sequence for a chromosome into a text file including the associated base pair sequence of every BED file.

We developed a convolutional neural network (CNN) to learn the mapping from the genotype to the phenotype, so the next step consisted of one-hot encoding the base pair sequence in the form of a

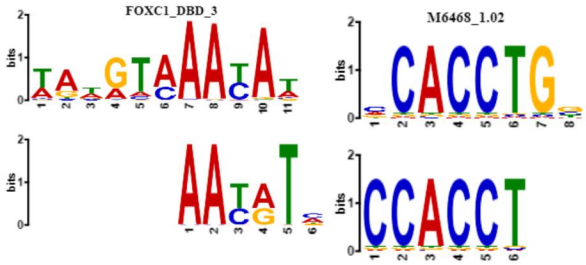

Figure 1: Example of genetic motif associated with blood cell differentiation (top), and the best 6 base pair sequence that activates our CNN's first layer filters (bottom). The match indicates that our model is learning.

Figure 2: Example of genetic motif in human CIS-BP found using the TomTom Motif Comparison Tool as with figure 1 (top) compared to the sequence that most activates the first layer of our best trained CNN (bottom).

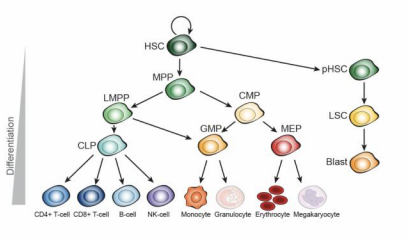

Figure 3: The cell lines that arise from hematic stem cells comprise the blood system of humans. Disruptions to this differentiation process result in diseases such as Leukemia. Our model predicts CA for each of these 18 cell types.

two-dimensional image with a narrow width of four. The dataset was significantly sparse at the outset, consisting of approximately 85% 0s for chromatin accessibility. To balance the distribution of output labels, we performed data augmentation by sampling only output labels with 1s as output.

4 Methods

Our model architecture consists of a convolutional neural network (CNN) that utilizes the Adam optimization technique to minimize a sigmoid cross-entropy with logits cost function given by:

$$J(Z) = \sum_i -y_i \log(\sigma(Z)) - (1 - y_i)\log(1 - \sigma(Z))$$

The input base pair sequences are one-hot encoded, yielding a dimension of mx4, while the output is an 18-dimensional vector representing the chromatin accessibility of each cell type (Figure 5). The model's general structure is presented in Figure 2, and consists of four total convolutional layers with intermediate layers implementing ReLU activation, batch normalization, and maxpool followed by two fully-connected layers and a sigmoid layer. Each filter consists of 50 different channels, with respective dimensions of (6x4), (32x1), (16x1), and (8x1) in addition to a uniform stride size of (1x1x1) and uniform max pooling dimensions of (1x4x1x1).

We introduced max pooling as a means of reducing the impact of location dependence of specific sub-sequences [15]. We found a significant improvement in dev set accuracy after introducing max pooling layers. The two sets of convolutional layers are followed by a flattening layer and two fully-connected layers, which reshape the intermediate outputs of the convolutional layers into (1x50) and (1x18). The structure of our neural network model architecture was heavily influenced by previous work completed by Kelly et al., who developed a similar model for deep learning in genomics with a similar CNN architecture[15].

After training our model for several iterations and saving the weights, we realized that our model was overfitting to the train set because our dev set error converged to the training set error after only 2-10 iterations. To regularize the weights used in the model, we introduced dropout, loss regularization, and data augmentation. We selected initial dropout hold rates of 0.9 for the convolutional layers and 0.9 for the fully-connected layers, but after a telescope search later decreased each of the two dropout rates to 0.8. After introducing a loss regularization term, our total sigmoid cross-entropy loss added an extra term:

$$J_{\text{regularization}} = \frac{\lambda}{2}||W||_2^2$$

To train our model more efficiently, we used an Amazon AWS Ubuntu p3.2xlarge GPU coupled with the NVidia CUDA library to parallelize our training process and make it more efficient. We found that

Figure 4: Example of training on the input to generate a genetic sequence that outputs all ones. The top sequence is the output of the training and consists of continuous values (just first 100 base pairs of length 1000 sequence). The bottom sequence identifies to the max value of each base pair to generate a one-hot encoding.

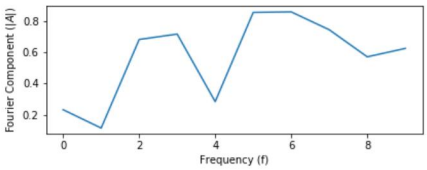

Figure 5: Fourier transform of the top sequence in figure 4 after averaging across base pairs. Peaks around 3 and 6 may indicate codon length and residual of the first layer filter size respectively.

Confusion Matrix: Prediction of (Erythrocyte, MEP, MPP) Chromatin Accessibility

	(0,0,0)	(1,0,0)	(0,1,0)	(1,1,0)	(0,0,1)	(1,0,1)	(0,1,1)	(1,1,1)
(0,0,0) Input	7082 (.90)	318 (.04)	0 (0)	216 (.03)	0 (0)	0 (0)	0 (0)	250 (.03)
(1,0,0) Input	768 (.76)	61 (.06)	0 (0)	53 (.05)	0 (0)	0 (0)	0 (0)	126 (.13)
(0,1,0) Input	170 (.66)	25 (.10)	0 (0)	20 (.08)	0 (0)	0 (0)	0 (0)	44 (.17)
(1,1,0) Input	88 (.58)	13 (.08)	0 (0)	12 (.08)	0 (0)	0 (0)	0 (0)	40 (.26)
(0,0,1) Input	30 (.63)	5 (.1)	0 (0)	2 (.04)	0 (0)	0 (0)	0 (0)	11 (.23)
(1,0,1) Input	15 (.39)	3 (.08)	0 (0)	1 (.03)	0 (0)	0 (0)	0 (0)	19 (.50)
(0,1,1) Input	66 (.48)	16 (.12)	0 (0)	11 (.08)	0 (0)	0 (0)	0 (0)	45 (.33)
(1,1,1) Input	61 (.12)	13 (.03)	0 (0)	24 (.05)	0 (0)	0 (0)	0 (0)	392 (.80)

Table 1: Confusion Matrix of three example cell types of the 18 total cells related to human blood. The three cell types, Erythrocyte, MEP, and MPP are all in a single lineage listed from youngest to oldest. The confusion matrix shows that the CNN only labels a 1 for a parent when the children are labeled a 1 as well. For example, column 8 shows the MPP only has a 1 when both MEP and Erythrocyte are 1 as well becaues MPP is the parent of MEP which is the parent of Erythrocyte (as seen in figure 3).

this greatly improved the speed of training and allowed us to attempt to optimize hyperparameters more efficiently using a manual telescope search.

From a biological perspective, the choice of a CNN architecture over RNN or another model was motivated by the biology of protein binding. During the expression of subsequences of DNA, proteins search out specific binding sequences called 'motifs.' Once found, the protein binds to the DNA and begins the process of expressing the connected DNA sequence as a protein. These proteins are allocated to different regions of the cell and result in different cell phenotypes.

We discovered that filters mimic this behavior by maximizing activation for base pair sequences that produce chromatin-binding proteins. After performing a Fourier decomposition on the length of base pair motifs, we noted that motifs with lengths of about 6 base pairs tended to bind to activation proteins with the highest frequency 5, which matches the size of our first convolutional layer.

5 Experiments/Results/Discussion

After training the model initially, we achieve an accuracy of 0.84 where accuracy is defined as the total fraction of output CA binaries that are predicted correctly. We average the confusion matrix across all 18 cell types to yield the results in table 5. Clearly, the data is skewed because there are many more zeros than ones, so we include auPRC as a better metric for our results than accuracy. These results are shown in 6 through a sensitivity and specificity curve. Although not perfect, we have begun learning the genetic sequences enough to show non-trivial sensitivity and specificity, which we maximize by choosing approximately 0.7 sensitivity and 0.7 specificity.

Although our results do not yet match those exemplified in "Basset" [15], we do have a model that has demonstrated powerful predictive capacities, with the potential to detect important genetic motifs and incorporate them into prediction of CA binaries. At this point, we begin to investigate

auxiliary methods for analyzing the trained structure of our model. We begin by finding cell motifs that maximize activation of the first layer of the model. Since the first filter has size 6 by 4, we can use a brute force method to check all 4^6 sequences of 6 base pairs to find the one that activates the filter most strongly. Specifically, we generate a random sequence of 6 base pairs, take its one-hot encoding, apply the first layer filter (and all channels of the filter), apply a linear rectifier, and sum the components. The genetic sequence that yields the highest value is deemed the genetic sequence that activates the filter most strongly. This approach searches for sequences that either strongly activate individual filters or that activate all the first layer filters simultaneously.

In attempting to analyze sequences that activate individual filters, we performed a double-blind study where the best sequence is found for a given first layer filter, then the TomTom tool is used to find matching genetic motifs in human CIS-BP. The best result is then matched with existing research publications to discover its relevance to hemopoeiatic stem cell differentiation. Due to the inherent subjectivity of online research, we perform the double-blind study using 50 genetic motifs, one from each filter and 50 randomly generated genetic motifs. For the 50 filter motifs, we find that 23 of them are related to blood cells, while 20 of the randomly generated sequences were related to blood cells. While we find 46% of our motifs match blood cells, which is on par with the $\approx 45\%$ found in "Basset,"([15]) our double-blind study reveals that the results are not yet statistically significant. Following this procedure, we found a bug in our data pre-processing that could explain the insignificance of the double-blind study. However, we note that to our knowledge Kelley et al. [15] do not perform a double-blind study in "Basset".

After using brute force to determine highest activation genetic sequences, we continue to generate full input sequences using the "training on input" method discussed in lecture [14]. Here we set Y to be all 1s because we want the input sequence X to contain all the important motifs. We first set X to be a trainable variable and fix all of our model weights. Next, we apply the same sigmoid cross-entropy loss function used during training of the model. This updates X directly, and we iterate until the cost is below a sufficient bound. Identifying specific motifs in the generated input is left to future work, but our preliminary results show that we have succeeded in creating an input that can be interpreted as a genetic sequence, as depicted in figure 4. Next, we seek to determine the rough structure of our generated input by taking the average across narrow dimension of four bases. This results in a 1-D series which we can analyze using a discrete Fourier transform. Figure 5 shows the results of this analysis which show rough peaks at 3 and 6, which could possibly be the characteristic length of a genetic codon and the residuals from the size of our first convolutional filter respectively. We leave confirmation of these results and further investigation of sequence structure to future work.

Lastly, we seek to uncover the cell differentiation structure of figure 3 by using revealing hierarchy within the CNN model. We find inspiration from Bilal et al. [19] who find class hierarchy through CNN's using different types of animal images. If our model understands the genetic mechanisms for CA correctly, it should have the information necessary to re-create the cellular hierarchy at in [19]. Instead of displaying the full confusion matrix, which would have all possible outputs (2^{18}), we choose three cell types that are within a single lineage. This ensures that our three cells have a clear hierarchy. The confusion matrix of these three, with 2^3 inputs and outputs, indeed, shows some sort of hierarchy. Namely, we find that if a parent cell is labeled, than our CNN labels all of the children as well. Table 1 shows the confusion matrix and revealed hierarchy.

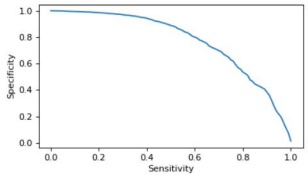

Figure 6: Sensitivity and specificity curve for our final CNN model using a test set of 1000 samples

Confusion Matrix: Prediction of all 18 Cell Types, Averaged

	0	1
0 Input	.735	.092
1 Input	.094	.079

6 Conclusion/Future Work

Future work would build on our methods of analyzing the model and improve the accuracy of our model. In our dataset we use only chromosome 1 - future work will compare this chromosome to the other 22. Likewise, subsequences of the data will be analyzed further to make generalization about what sequences are most sensitive to changes in differentiation. After training the model to higher accuracy using better hyperparameters, we will obtain more accurate confusion matrices that can yield potential reconstruct the entire blood cell hierarchy. We also plan to train on the input more by comparing generated inputs that yield different CA outputs.

After achieve reasonable accuracy for predicting CA, we analyzing the model using three distinct methods, each of which lends credence to our model and opens the door for novel exploration of the model in the future.

7 Contributions

Alex: I preprocessed the data, configured the GPU to run our model with Amazon AWS, met with Surag to discuss applicable research, processed the results of the base pair matches by matching to o

Brandon: I read in the data from a raw file that Alex generated, created the model in tensorflow and spent many days fine tuning the architecture, added dropout, regularization, and methods for saving and loading parameters. After running the model to reasonable accuracy on my CPU over many days, I came up with three methods for analyzing the model. These are the three models discussed in the discussion section of the paper. I also generated the sensitivity and specificity curves, defined the test metric to be auPRC, wrote sections 2, 5, and 6 of the paper and generated all the plots used for section 5.

Our code can be found at: https://github.com/Alekxos/dropthebasepairs/upload/master

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[4] Shobhit Gupta, JA Stamatoyannopolous, Timothy Bailey and William Stafford Noble, "Quantifying similarity between motifs", Genome Biology, 8(2):R24, 2007.

[5] Kelly et al. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Research. 2015.

[6] Hindorff LA, Sethupathy P, Junkins HA, Ramos EM, Mehta JP, Collins FS, Manolio TA. 2009.Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. Proc Natl Acad Sci 106: 9362–9367.

[7] Peyton Greenspan; graduate student, Stanford Biology Department.

[8] CS 230; Coursera. Stanford University; Winter 2018. Andrew Ng.

[9] Chromatin accessibility: a window into the genome. Tsompana and Buck. Epigenetics & Chromatin. 2014.

[10] Assembly of the Human Genome. UCSC. http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/

[11] bedtools: a powerful toolset for genomc arithmetic. Quinlab Lab. University of Utah. http://bedtools.readthedocs.io/en/latest/

[12] Alipanahi, B. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology, 33(8), 3–1. https://doi.org/10.1038/nbt.3300

[13] Beer, M. A., Tavazoie, S. (2004). Predicting Gene Expression from Sequence. Cell, 117(2), 185–198. https://doi.org/10.1016/S0092-8674(04)00304-6

[14] Katanforoosh, K., Ng, A., Mourri, Y. B. (2018). CS230: Lecture 5 Attacking Networks with Adversarial Examples -Generative Adversarial Networks. Retrieved from http://cs230.stanford.edu/files/lecture-notes-5.pdf

[15] Kelley, D. R., Snoek, J., Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Research, 26(7), 990–9. https://doi.org/10.1101/gr.200535.115

[16] Mahendran, A., Vedaldi, A., Schmid, C. B. (2016). Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. Int J Comput Vis, 120, 233–255. https://doi.org/10.1007/s11263-016-0911-8

[17] Omatsu, Y., Seike, M., Sugiyama, T., Kume, T., Nagasawa, T. (2014). Foxc1 is a critical regulator of haematopoietic stem/ progenitor cell niche formation. https://doi.org/10.1038/nature13071

[18] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2013). Intriguing properties of neural networks. Retrieved from http://arxiv.org/abs/1312.6199

[19] Bilal, A., Jourabloo, A., Ye, M., Liu, X., Ren, L. (2018). Do Convolutional Neural Networks Learn Class Hierarchy? IEEE Transactions on Visualization and Computer Graphics, 24(1), 152–162. https://doi.org/10.1109/TVCG.2017.2744683

[20] Timothy L. Bailey, Mikael Bodén, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, William S. Noble, "MEME SUITE: tools for motif discovery and searching", Nucleic Acids Research, 37:W202-W208, 2009.

```
TGTGGTCTTCATCTGCAGGTGTCTGACTTC
CAGCAACTGCTGGCCTGTGCCAGGGTGCA
AGCTGAGCACTGGAGTGGAGTTTTCCTGT
GGAGAGGAGCCATGCCTAGAGTGGGATT
CGAGGCTCTAGCCATATCCTGGCTAGGACT
```

Figure 7: Sample base pair sequence based on Chromosome I; one-hot encoded before use in training set.

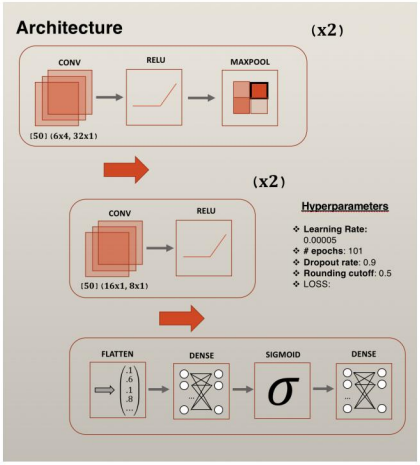

Figure 8: Architecture of the convolutional neural network. Consists of four convolutional layers followed by two densely-connected layers.