Application of Convolutional Neural Networks in classification of Cis-Regulatory elements with ATAC-seq data: The Preliminary Steps

Jia Shi
Department of Data Science

Ziyang Zhang
Department of Bioengineering

Abstract

To better understand the mechanisms of gene expression and subsequent phenotypic divergence, it is critical to further investigate the non-coding region of the genome, which accounts for 99% of DNA [1]. In this paper, we adopt a variety of Convolutional Neural Networks (CNNs) designed to classify 500-base-pair sequences centered at peaks into two major Cis-Regulatory Elements(CREs) classes: Promoter or Represser. We first train a CNN using solely V-plots [2] generated by plotting fragment length against coordinate offsets from the peak center. Then, we build a CNN that classifies the sequence based on convolution kernels inspired by Position Weight Matrix (PWM). Finally, we explore the possibility of combining multiple CNN models' output to further enhance performance in CRE identification. While this paper is far from accomplishing a generalized model for CRE identification, it does serve as a robust first step in proving the potential of CNN fueled by ATAC-seq data.

1. Introduction

1.1 Human Genome and CRE

The modern understanding of the human genome was pioneered by the Human Genome Project that "catalogued a parts list of most human genes" [3]. To decipher the proteins and functions of these "parts", many subsidiary projects were initiated. A prime example was the ENCODE (Encyclopedia of DNA Elements) Project that aimed to understand the functional elements of the human genome. While we have made significant progress in understanding the coding regions of DNA, deciphering the non-coding region proves to be a more challenging task , both quantitatively (99% of genomes are non-coding regions) and qualitatively (complexity and diversity of molecular interactions).

One of the most important category of sequences in non-coding DNA is cis-regulatory elements (CREs) what moderates the binding sites for other molecules and hence facilitate the activation and sustainment of gene expression. Current hypothesis in academia is that the mutations in CREs are now considered the “most prevalent cause of phenotypic (especially morphological) divergence” [1]. Correct labelling of CREs is the first step in proving or disproving that hypothesis.

1.2 Previous Attempts in labelling genome

There have been various attempts to label the non-coding region of the human genome. Probabilistic models, such as Hidden Markov Models (HMMs), were among the first to predict genes based on canonical data such as transcriptional, translational and splicing signals. The pioneer of such HMMs was GENSCAN that debuted in the late twentieth century [5]. Such models showed promises in predicting introns and exons in the human genome. More recent advances in HMMs were able to classify “amino acids into structural classes” [4] and identify “regions or states enriched in specific combinations of histone modifications” [6]. Both of these models were competent in demonstrating the practicality of genome labelling when sufficient signal data were provided.

With the advent of deep learning, numerous models were developed to tackle this issue. CNNs are tested robustly with various architectures and hyperparameters to predict DNA-protein binding with sequences as inputs [7]. Inspired, recent researchers expanded the use of CNN to other areas such as prediction of enhancer-promoter interactions [8]. These models mainly use data generated by ChiP-seq and RNA seq.

1.3 Breakthrough of ATAC-seq

Assay of Transposase Accessible Chromatin sequencing (ATAC-seq) is a major technological breakthrough in bioinformatics. The enzyme used by ATAC-seq, Tn5 transposase, is hypersensitive to open chromatin regions and requires 500-50,000 cells [9] for the entire protocol. As a result, ATAC-seq has specificity comparable to canonical methods while is superior because it doesn’t require “rigorous library selection” [9]. Such versatility is critical for understanding of humans’ and other mammalian species' epigenomes. We are therefore experimenting with CNN using ATAC-seq data in this project.

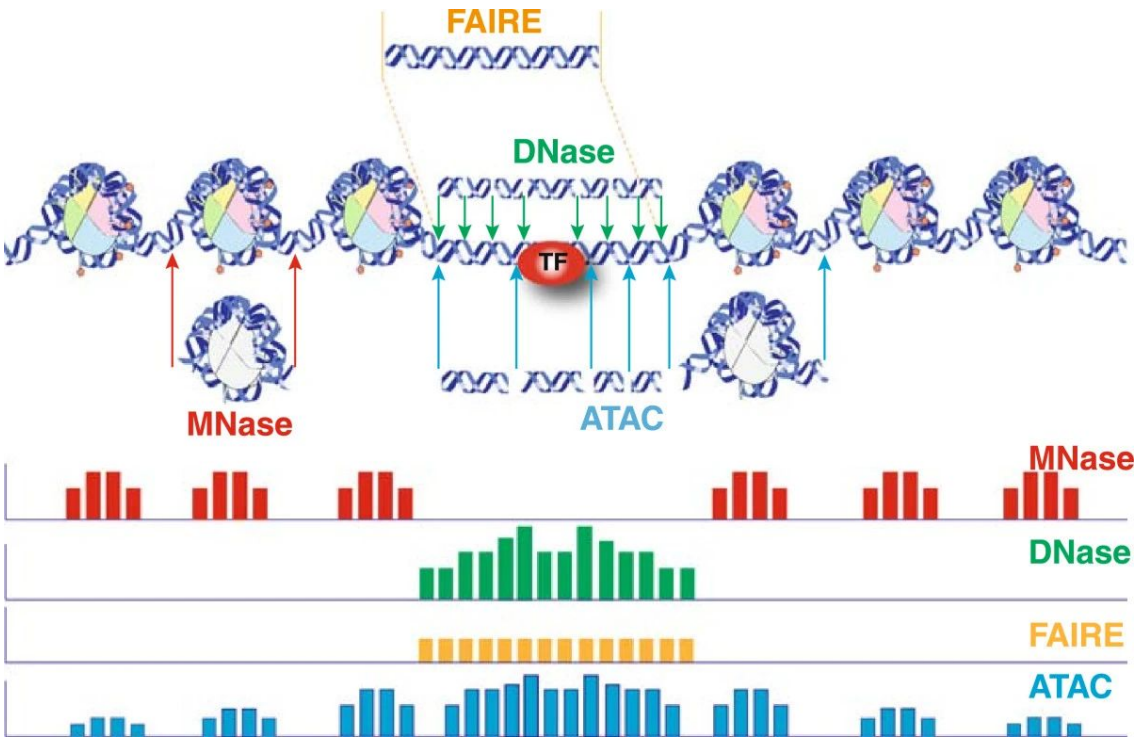


Figure 1*: Schematic comparison of ATAC-seq against previous sequencing methods. ATAC-seq is superior in its versatility and sensitivity.*

1.4 Problem Statement

The ideal model would be able to use both raw and preprocessed ATAC-seq data as input and classify peaks as one of the known CRE.

1.5 Main Deliverables

While the ideal model would certainly aid our understanding of genomic interactions on a molecular level, we must first prove its validity by experimenting with the following CNNs:

1. A simple CNN that predicts CREs based solely on Vplot images as input.
2. A Simple CNN that predicts CREs by scanning sequences for motifs.
3. An Ensemble CNN that predicts CREs by incorporating input from the previous two CNNs.
4. Additional benchmarking to find the most suitable combination of architecture and hyperparameters.

2. Dataset

2.1 Preprocessing

The data used in this project is Single-Cell ATAC-seq produced for the Mouse Cis-element Atlas project (CATlas) [10]. Standard bioinformatic preprocessing is performed by experienced bioinformatician using industry standard tools. In particular, the peak sequences are identified using MACS and cropped into 500 bp sub-sequences around the peak [11]. ATAC-seq are generated using standard alignment tools and stored in a pari-end bed file.

2.2 Vplots

Vplots are scatter plots of fragment length vs coordinate offset from center of fragment to center of peak. The region of sequences that promotes expression of genes (in our case Active Promoters) should exhibit a "V" pattern whereas the region of sequences that suppresses expression of genes should not. The aggregate V plots of typical sequences are demonstrated in figure 1 and distribution of fragment length are shown in histograms in figure 2.

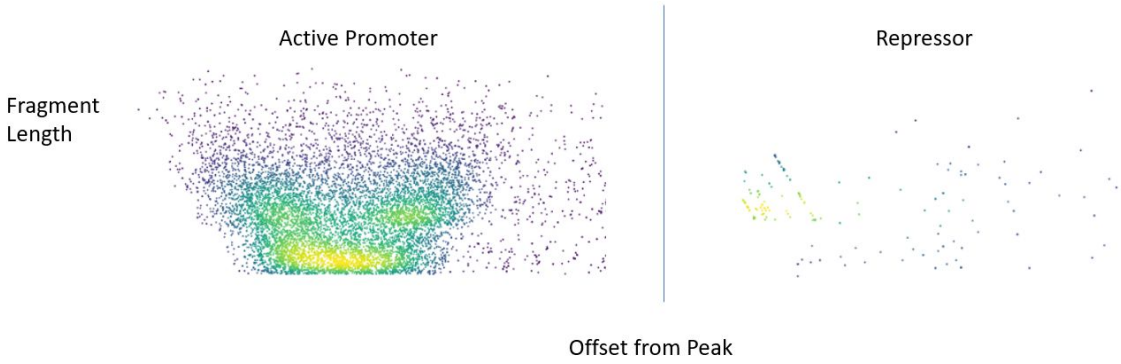


Figure 1*: Aggregate Vplot for pro-expression class (left) and anti-expresion class (right). Left plot exhibits both more fragments within the peak region and a distinct V-shape. Right plot is much more scattered and does not exhibit a distinct pattern. Lighter color indicates higher concentration of points.*

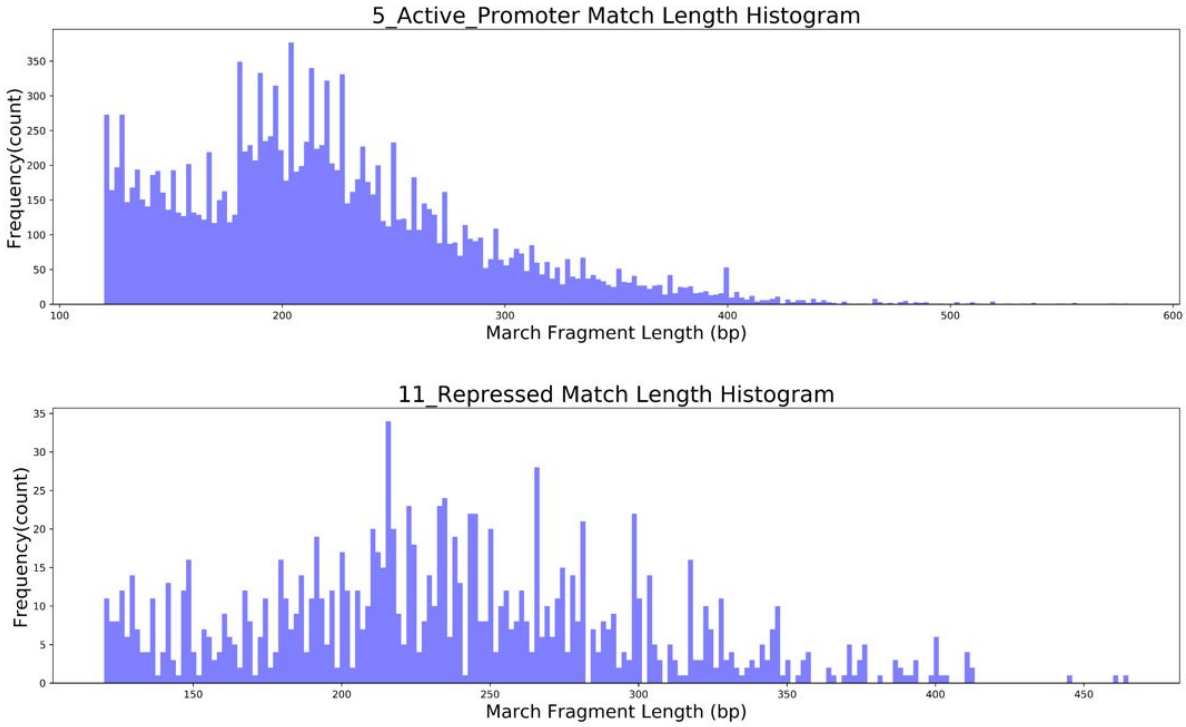


***Figure 2**: Histograms of fragment length. Top plot's (active promoter) fragments length are concentrated on the shorter reads whereas bottom plot's (repressor) is more evenly distributed.*

2.3 Generation of Classification Labels

All peaks are labelled if they overlap with labelled genome regions proposed by HMM by at least 50%. For the purpose of this pioneer project, only two out of six total classes are selected due to computational limitations.

2.4 Dataset Summary

For the repressor class, a total of 2497 peak segments of length 500 sequences are extracted.
For the active promoter class, a total of 2221 peak segments of length 500 sequences are extracted.
Vplots are drawn on these peaks as previously described.

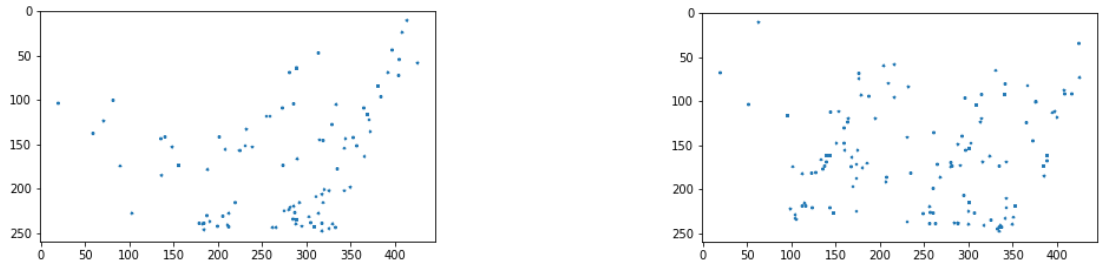
3. Network & Result

To handle the two different types of data, we propose two different CNNs: Vplot CNN and Motif CNN. We have conducted extensive comparative studies on different architectures in classifying two different classes of genes. During training of the models, we are using a learning rate of 1e-3, and with cosine scheduler of learning rate decay. We use Adam as our optimizer and initial the model weight randomly. For models like ResNet and MobileNet, we didn't use a pre-train state dict but chose to start training from zero.

It's worth mentioning that the upper bounding of the validation accuracy is 85% because of the labeling error and the possible pattern error in the original dataset. WIth training, most of our network have achieved slightly more than 70% of validation accuracy.

3.1 Vplot Model

The two plots below are the training data of different classes. As shown in figure 3, it's very hard to tell their difference even with human eyes, thus it's very noisy in training.



class 0 class 1

Figure 3*: Sample Vplot around single peak. Due to noise in data, the plot for class 0 (left) and class 1 (right) does not differ significantly.*

For the image-based classification task, the result of our comparison studies is shown below. Each boxplot was measuring the validation accuracy after each epoch of 12 in total.

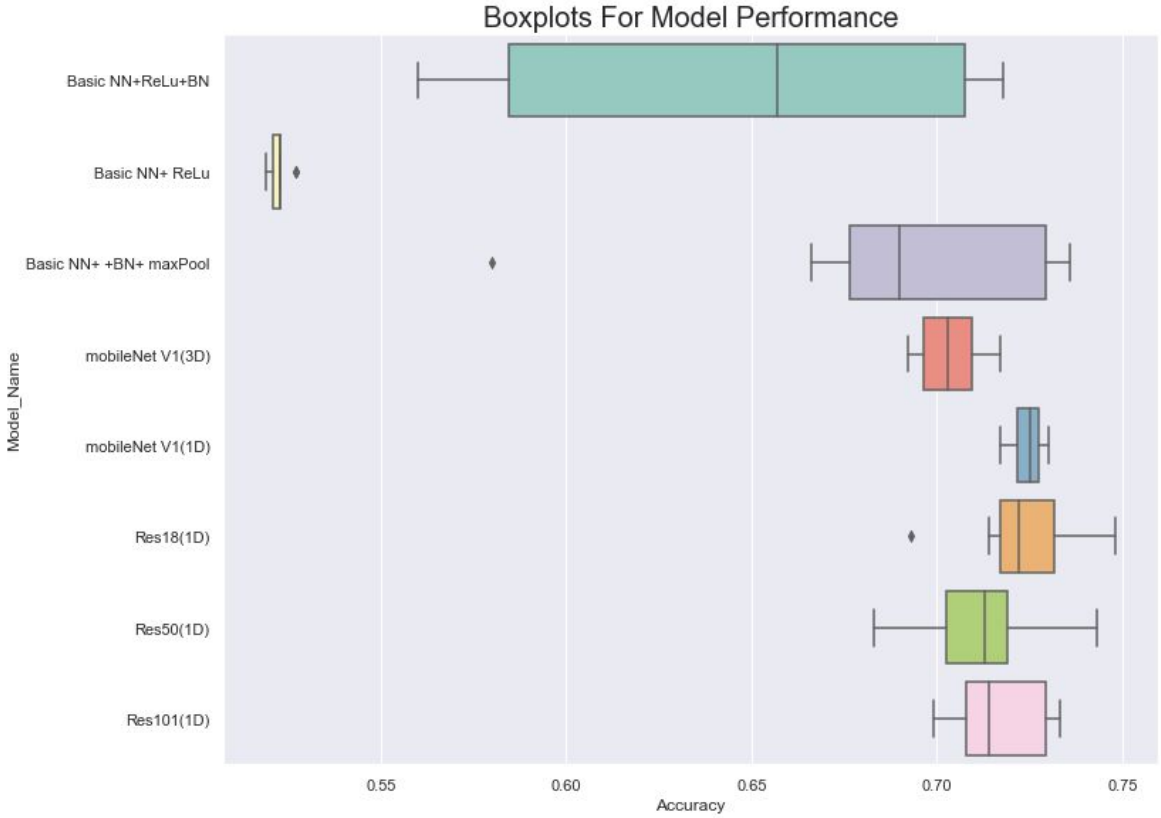


Figure 4*: Boxplot for accuracy/performance of different NNs and CNNs tested.*

As we can see in figure 4, we had tried classifying with pure Fully-Connection layers, and compared with the addition of different layers, like pooling. Also, we had tried to use some CNN-based models like MobileNet V1 and ResNet families.

3.1.1 Basic NN+ReLu & Basic NN+ReLu + BN

When we were only using pure FC layers with activation function of ReLu, the loss doesn't seem to decrease much under gradient descent. However, when we add a Batch normalization after each FC layer, the loss starts to reduce and we can reach around 73% evaluation accuracy eventually. However, it's still not robust enough as shown on the graph. It may take a while to reach the final accuracy. The data is very noisy, since there's no specific features that correspond for the prediction. And the distribution of data may shrink from batch to batch which makes it harder for NN to capture features for classification. Batch Normalization reduces this covariance shrift among batches which accelerate the training speed and allows networks to learn features more efficiently.

3.1.2 Basic NN+ReLu + BN & Basic NN+ReLu + BN + Max-Pool

As mentioned above, the image data is noisy, especially on a local scale. The addition of max pool layer allows the model to include global reasoning as well as reduce the local noise by extracting the most nominet local features, which make it more robust in training and the model will converge in a faster speed as well. With Max pool layers, we can reach a final accuracy of 74% and can reach relatively high accuracy within a few epochs compared to models without max pool layer.

3.1.3 3D & 1D data

The generated images in the dataset have 3 channels, as the regular RBG images. However, the information stored is purely binary, which can be represented by grayscale images. Thus, we had trained a MobileNet V1 for comparing data with 3D and only 1D grayscale. And as shown in the images, the 1D grayscale model has a slightly higher accuracy and more robustness. This is because 3D data have repeated information in each channel which requires more time and more complex structure to encode, thus 1D data performs slightly better.

3.1.3 Networks & deeper networks

We are interested in whether more layers would yield better performance. To investigate this hypothesis, we compare different networks in the ResNet family. As shown in figure 4, the performance of ResNet 18 is better than both of ResNet 50 and ResNet 101. Since the information in the images is very limited (only binary information), a deeper network would remove more local information along the training and thus the result would be slightly worse compared to shallower networks. And also it's much slower to train a deeper network.

3.2 Motif Model

In this section, we compared models with different number of header of the encoder, different number of conv layers, and specific layer like dilated conv layer.

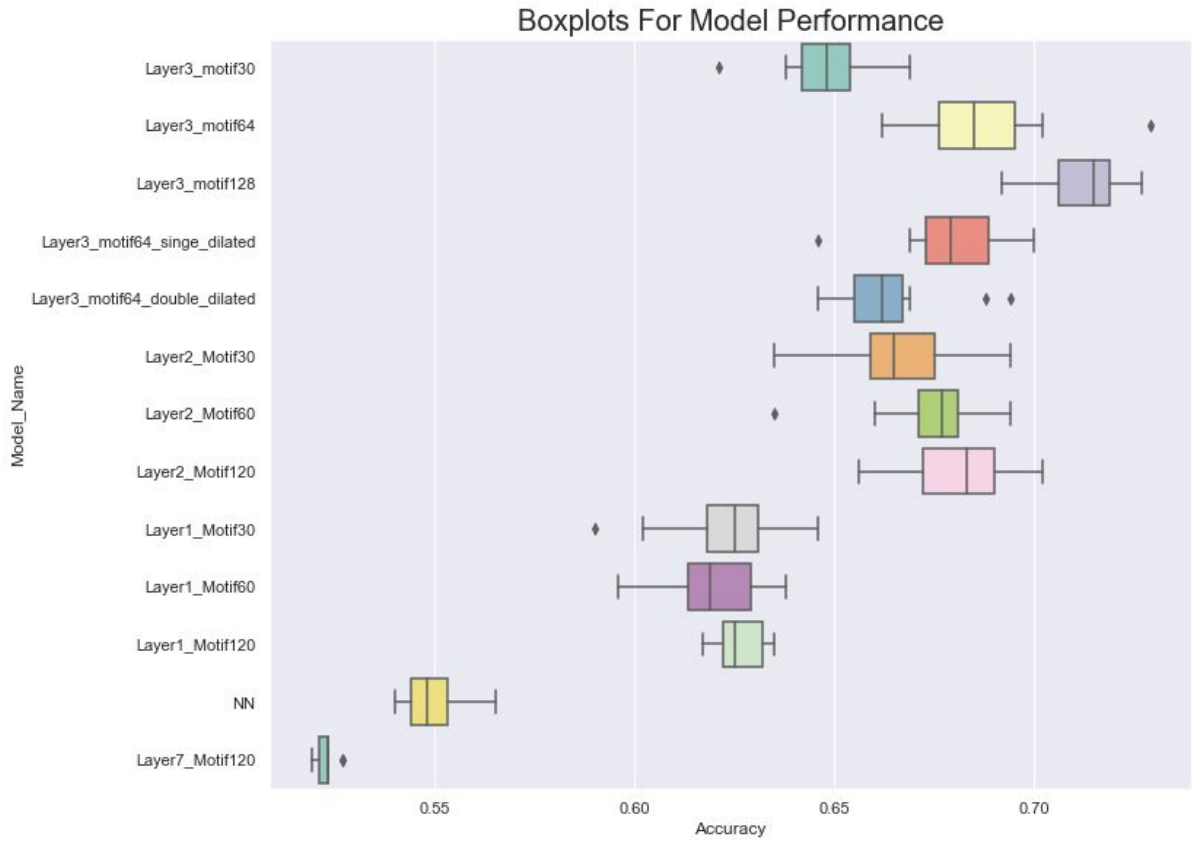


Figure 5*: Boxplot for different Motif CNNs and NNs tested. Layer(number) refers to the number of convoluted layers in the model. Motif(number) Refers to the number of parallel encoders connected to an output feature map. (Single/Double) dilated layer refers to the number of dilated layers in the network. For instance,* ***Layer3_motif30*** *refers to a network with 3 convolutional layers and 30 parallel encoders.*

3.2.1 Dilated Layers

We included Dilated Convolution Layers in our study inspired by [3]. Theoretically, Dilated Conv can increase the size of the receptive field without reducing the size of the feature map, which would capture more global information without loss of the spatial information. However, as shown in figure 5, the Dilated Layers do not perform better compared to networks with regular convolution and pooling layers. This is potentially because the Dilated convolution layer would also lose some local information with holes in the convolution.

3.2.1 Number of headers

We tried to compare models with different numbers of the encoder header. As shown in the graph above, the increase of the number of headers does increase the validation accuracy. This is because a multi-header encoder would capture information from multiple scale and multi-length patterns, which corresponds to the variety of DNA patterns in the data sequence.

3.2.3 Network & deeper network

Similar to above, we had conducted an experiment of networks of different depth. As shown in figure 5, the networks with 7 convolution layers perform much worse than networks with only 2 or 3 convolution layers. This is because the increased number of cov layers and pooling would remove local information along the way.

3.3 Ensemble CNN

We had also tried to ensemble the models of both image-base and sequence-based networks. For each participating model, we store the data after the softmax layer, which is the possibility of prediction for two different classes. Then we sum them up and output the class with higher value. We had ensemble an image-base model of 70% validation accuracy, and a sequence base model of 65% validation accuracy. The ensemble model reached 73% validation accuracy.. We tried to ensemble an image-based model of 74% and accuracy and a sequence-based model of 74%, yielding a model with 76% of validation accuracy.

3.4 Summary

Most of our network can reach around 70% of validation accuracy and 99.99% of training accuracy. The highest accuracy reached is by the ensemble model, peaking at 76.5% validation accuracy. The highest possible accuracy we can get is 85%(the upper bound of the dataset), which indicates the potential to reach around 90% validation accuracy if we have maped it into a 100% scale.

4. Discussion

4.1 Strength and limitations of vplots

Generally speaking, the ATAC-seq reveals information about the open chromatin regions. As previously shown in aggregated Vplots, around peaks that serve as accessible binding sites, the concentrated shorter fragments correspond to nucleosome-free fragments. On the other hand, for the peaks which are not accessible, the fragments are much less concentrated and their lengths

are evenly distributed. These characteristics allow the model to perform classification of different peaks.

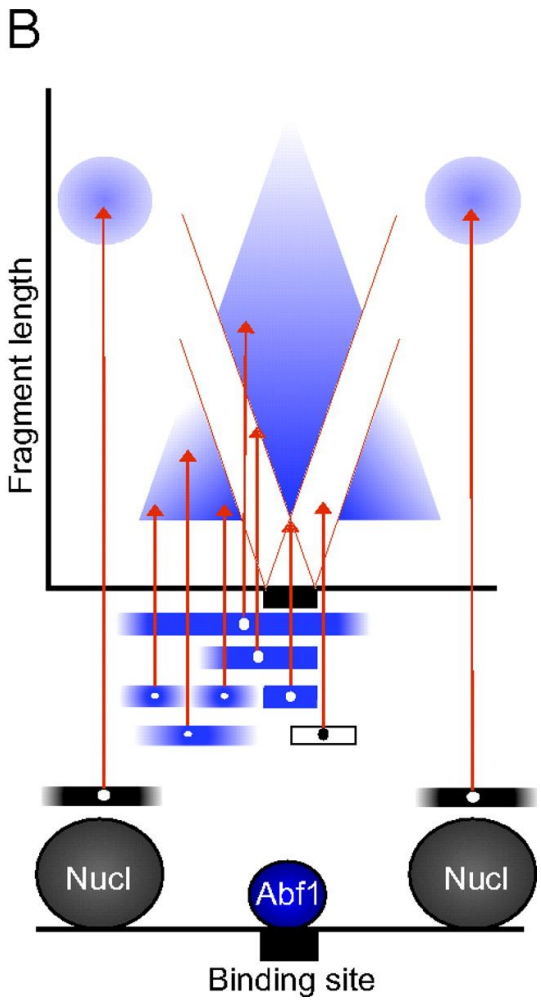


Figure 6: *Schematic representation of Vplot. The shorter fragments (blue) refer to the nucleosome-free fragments while longer fragments correspond to closed heterochromatin fragments (black) [2].*

However, the limitations of the vplots approach is also significant. Firstly, due to the inherent noise to biological data, vplot around individual peaks are extremely noisy. Despite using numerous de-noising methods, some of the produced vplots still fail to distinguish themselves from rivaling classes. Secondly, Vplots in inherently loses the nuance in differentiating classes exhibiting similar chromatin accessibility. Therefore, Vplots alone could not provide enough information to train a comprehensive model.

4.2 Convolutional layers in Motif CNN

The general performance trend of Motif CNN carries biological interpretation. With an increased number of parallel convolution layers, the prediction accuracy increases. Such improvement could be attributed to better capture of motif embedding. Consider each filter as a scanner encoded by a certain motif, each filter is therefore checking whether this motif is present in the sequence. Naturally, an increased number of such filters expands the model's ability to examine more motifs. We also ran motif analysis using Homer, an industry standard software, and discovered over 40 significant motifs (p-value < 1e-12). This further explains why increased number of convolutional filters enhances model performance.

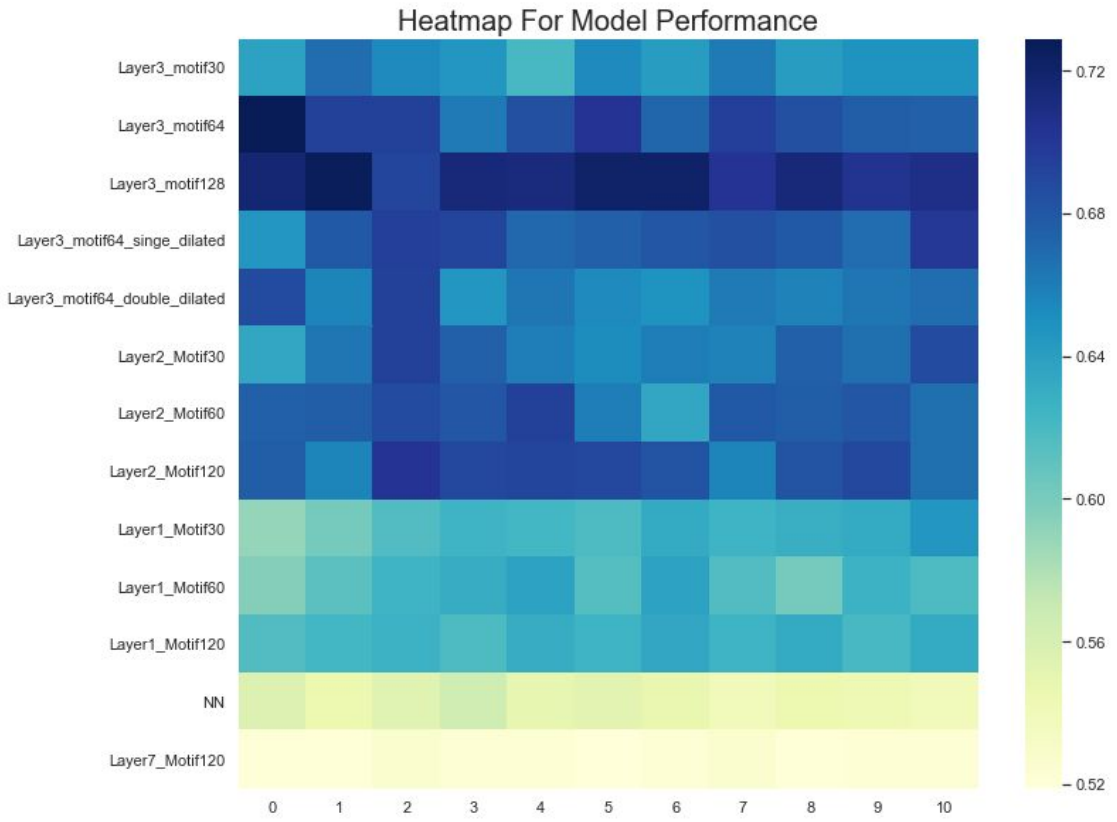


***Figure** 7: Heatmap for Motif CNN performance. Darker color indicates higher accuracy in the validation set.*

Additionally, increased number of convolutional layers after the initial motif scanning also improves validation accuracy. This could be explained by the presence of higher order structure where certain combinations of motifs are indicative of a particular class. However, simply increasing the number of convolution layers does not guarantee better performance. In fact, in our experiment this scheme yields the lowest of all models. Although somewhat counterintuitive, this reveals the inevitable tradeoff between capturing higher order structure and losing local context. This abysmal performance of our 7-layer convolutional model indicates that losing too much context fails to make accurate predictions.

Lastly, the performance simple feedforward neural network (NN) demonstrates the necessity of parallel motif-scanning convolutional layers. The information in the sequence could not be properly extracted in the absence of such layers.

5. Future Considerations.

It is evident that our proposed model could perform the binary classification with decent accuracy with much space for further improvements as discussed below:

1. Experiment with Sequence-based Recurrent Neural Network which takes into account the sequential context present in the genome.
2. More sophisticated preprocessing to make Vplots among classes more differentiable.
3. Incorporate additional information (e.g. RNA-seq) to broaden depth of data the model could learn.

References

[1] Wittkopp, P., Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. Nat Rev Genet 13, 59–69 (2012). https://doi.org/10.1038/nrg3095

[2] Henikoff, Jorja G et al. “Epigenome characterization at single base-pair resolution.” Proceedings of the National Academy of Sciences of the United States of America vol. 108,45 (2011): 18318-23. doi:10.1073/pnas.1110731108

[3] Hood, Leroy, and Lee Rowen. “The Human Genome Project: big science transforms biology and medicine.” Genome medicine vol. 5,9 79. 13 Sep. 2013, doi:10.1186/gm483

[4] Blasiak, Sam & Rangwala, Huzefa. (2011). A Hidden Markov Model Variant for Sequence Classification.. IJCAI International Joint Conference on Artificial Intelligence. 1192-1197. 10.5591/978-1-57735-516-8/IJCAI11-203.

[5] Burge, Christopher; Karlin, Samuel (1997). "Prediction of complete gene structures in human genomic DNA" (PDF). Journal of Molecular Biology. 268 (1): 78–94. CiteSeerX 10.1.1.115.3107. doi:10.1006/jmbi.1997.0951. PMID 9149143. Archived from the original (PDF) on 2015-06-20.

[6] Gireesh K. Bogu, Pedro Vizán, Lawrence W. Stanton, Miguel Beato, Luciano Di Croce, Marc A. Marti-Renom Molecular and Cellular Biology Feb 2016, 36 (5) 809-819; DOI: 10.1128/MCB.00955-15

[7] Haoyang Zeng, Matthew D. Edwards, Ge Liu, David K. Gifford, Convolutional neural network architectures for predicting DNA–protein binding, Bioinformatics, Volume 32, Issue 12, 15 June 2016, Pages i121–i127, https://doi.org/10.1093/bioinformatics/btw255

[8] Zhong Zhuang, Xiaotong Shen, Wei Pan, A simple convolutional neural network for prediction of enhancer–promoter interactions with DNA sequence data, Bioinformatics, Volume 35, Issue 17, 1 September 2019, Pages 2899–2906, https://doi.org/10.1093/bioinformatics/bty1050

[9] Yan, F., Powell, D.R., Curtis, D.J. *et al.* From reads to insight: a hitchhiker’s guide to ATAC-seq data analysis. *Genome Biol* 21, 22 (2020). https://doi.org/10.1186/s13059-020-1929-3

[10] Yang Eric Li*, Sebastian Preissl*, Xiaomeng Hou, Ziyang Zhang, Kai Zhang, Rongxin Fang, Yunjiang Qiu, Olivier Poirion, Bin Li, Yiming Yan, Hanqing Liu, Xinxin Wang, Jee Yun Han, Jacinta Lucero, Samantha Kuan, David Gorkin, Michael Nunn, Eran A Mukamel, M. Margarita Behrens, Joseph R. Ecker, Bing Ren. An Atlas of Gene Regulatory Elements in Adult Mouse Cerebrum. bioRxiv. doi: https://doi.org/10.1101/2020.05.10.087585

[11] Zhang, Y., Liu, T., Meyer, C.A. et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol 9, R137 (2008). https://doi.org/10.1186/gb-2008-9-9-r137

Additional reference used in coding

[12] Correct Way To Parse A Fasta File In Python
[13] sklearn.preprocessing.LabelEncoder — scikit-learn 0.23.1 documentation
[14] MobileNet
[15] seaborn.boxplot — seaborn 0.10.1 documentation
[16] seaborn.heatmap — seaborn 0.10.1 documentation
[17] How can I make a scatter plot colored by density in matplotlib?