Deep Neural Networks trained on DNA sequences to identify mutations that lead to Amyotrophic Lateral Sclerosis (ALS).

Aditya Vardhan Josyula
Masters Research Thesis
Artificial Intelligence
26 – 04 – 2021

Abstract

Amyotrophic lateral sclerosis (ALS) is a terminal disease whose onset may largely be determine by mutations in the non-coded region of deoxyribose nucleic acid (DNA). These mutations disrupt the transcription factors resulting in aberrant regulation of gene expression within the motor neurons resulting in neuro muscular degeneration. At the root of such mutations in DNA lie motif (Lanchatin, 2017), which are short conserved sub-sequences within DNA sequence in which mutations play a key role in regulating transcription.

In this project, we build a box of motifs using deep learning which can identify the active DNA sequences that comprise of damage causing mutations. I build two deep learning networks. (1) Convolutional Neural Network (CNN) and (2) Hybrid model which is a combination of CNN and long short-term memory (LSTM). These architectures are trained on active motif reference sequences and inactive reference DNA sequences in the non-coding region of the DNA extracted from human reference genome.

To determine the efficiency of the deep learning models in identifying mutations, I train the model architectures on blood lymphoblastoid. Mutations in blood lymphoblastoid are known to effect transcription. Next, we zoom in on the same region and train the models on regions around transcription start sit (TSS). These are regions where the mutations typically have strongest effect since these are sites where the process of transcription is initiated. To evaluate the model performance, I use a test set that comprises of (1) Genotype tissue expression (GTEx) which comprises of some motifs that could effect transcription as observed in people. Transcription is a process in which DNA gets converted into protein. Disrupting transcription leads to aberrant protein synthesis. These motifs are derived using traditional standard framework such as expression quantitative trait loci (eQTL) which comprise a list of effects of certain mutations across species which are known to affect a single cell gene expression. (2) Project MinE data consists of observed motifs in patients and controls in which some mutations may disrupt transcription leading to aberrant protein regulation which ultimately leads to ALS.

Testing both the model architectures on GTEx and MinE shows the reliability of deep neural networks in identifying motif mutations which are likely to disrupt transcription. Having determined the performance of the deep learning models on lymphoblastoid, we test the efficiency of the models in identifying ALS mutations by training them on non-coding DNA sequences intrinsic to complex neuropsychiatric diseases from lower motor neuron and test the models on Project MinE data.

Although previous deep learning models trained on motifs (Yue & Wang, 2018; Beer & Tavazoie, 2004; Alipanahi et al, 2015; Salekin & Zhang, 2017) show some success in predicting significant mutations that affect gene expression, we see in this project that both the models underperform in predicting significant mutations on the imbalanced GTEx and MinE datasets. The CNN model trained on blood has an average area under curve (AUC) of 0.42. The average AUC of the hybrid model on blood is 0.41. Similarly, the F1 score of the CNN on trained on blood is 0.07 and the F1 score of the hybrid model trained blood is also 0.07. The low AUC and F1 values show underperformance by the model. The CNN and hybrid models trained on lower motor neuron predict 12.39% and 6.40% of the active mutations in Project MinE.

***

Table of Contents

***

1. Introduction

ALS is terminal neurodegenerative disease which is characterized by degeneration of upper and lower motor neurons (Hardiman et al, 2017). The upper motor neurons are projected from the cortex to the brain stem and spinal cord. The lower motor neurons on the other hand are projected from the brain stem or the spinal cord to the muscles. The neuropathological hallmark of ALS is *protein inclusion* (Kierman & Vucic, 2011; Hardiman et al, 2017). Protein inclusion is the aggregation of proteins in a cell body. This significant hallmark is observed across several neurodegenerative disease.

ALS can either be classified as sporadic or familial. *Sporadic ALS (SALS)* applies when there is no known history of other family members with the disease. In contrast, *familial ALS (FALS)* applies when there is more than one occurrence of disease in the family. Earlier research in the field had only suggested 5-10% of ALS cases as familial (Byrne & Walsh, 2011), meaning it arises in families with a history of ALS.

The primary symptoms observed across ALS patients are associated with motor dysfunction which includes muscle weakness, paralysis, and discomfort in swallowing. In some patients, the degeneration extends to frontal and anterior temporal lobes, damaging the executive network of the brain leading to cognitive impairment as observed in 50%, behavioural change and *Fronto Temporal Dementia (FTD)* as observed in 13% of the patients. These general symptoms observed across ALS patients, along with the identification of specific rare genetic variants have contributed to re-characterising ALS as a progressive neurodegenerative disease.

Each human comprises of a DNA. The basic biological process involved in the survival of humans is that a DNA gets converted into a functional product protein (Leavitte, 2004). This happens in two steps. A DNA first gets transcribed into intermediate product which stores information regarding the amount of protein that needs to be produced. The process in which transcription occurs is called gene expression. In the second step, the intermediate product is converted into protein. There are two kinds of DNA. First, a coding DNA gets transcribed and translated resulting in adequate amount of functional product (protein). Second, a non-coding DNA only gets transcribed and not translated. However, some non-coding DNA sequences disrupt the transcription leading to aberrant protein synthesis in the later steps. In our body, 99% of the DNA is non-coding and most of the disease-causing mutations occur in the non-coding DNA (Brown, 2017; Hardiman et al, 2017).

How is the transcription process initiated? There are proteins called transcription factors (TF) which when bind with the DNA, the process of transcription is initiated. However, how does the DNA sequence know that a certain TF is compatible? There are motifs within a DNA, which are a single nucleotide subsequence of a DNA sequence. The motif mutation determines the affinity with which a certain TF binds with the DNA sequence initiating the process of transcription. In coding DNA this further leads to adequate amount of protein synthesis. Although some non-coding DNA find their compatible TFs making the sequence transcription ready, some motif mutations also disrupt the TF and their corresponding transcription factor binding site (TFBS) leading to aberrant protein synthesis in the surrounding regions.

This pattern of a motif mutation in non-coding DNA disrupting the TF and their corresponding TFBS is seen among ALS patients. This aberrant protein regulation is a

hallmark of ALS. Therefore, we build a deep learning tool to identify which motif mutations among ALS patients are likely to disrupt the TFBS leading to aberrant protein synthesis. Deep learning is a subset of Artificial Intelligence which uses multi-layered neural networks which replicates the connections of neurons in the brain. A deep learning approach uses a training set to learn the complexities involved. The learning of the model is finally tested using a test set. Typically, the train-test split is in a ratio of 70% - 30%. The test set comprises of two labels, a true label and a false label which help in evaluating the performance of the model on the test set.

In this work, we build two deep learning models based on the successful performance in identifying motif mutations of the already existing architectures. We will discuss the model architectures in more detail in the upcoming sections. To determine the efficiency of the models in the identifying motif mutations, we first train the models on non-coding DNA sequences from blood lymphoblastoid. We establish the efficiency of the model by judging its ability to classify motif mutations that effect transcription using genotype tissue expression (GTEx; Londsale et al, 2013) from the mutations that disrupt the TF and their corresponding TFBS using Project MinE (Project MinE ALS Sequencing consortium, 2018). The MinE data set comprises of mutations as observed in ALS patients. Having established the efficiency of the deep learning models in identifying motif mutations in blood, we next train our model on non-coding DNA sequences from lower motor neuron to identify the likely motif mutations that could disrupt the TF leading to ALS.

2. Background

This section will include some basic biology concepts such as deoxyribonucleic acid (DNA), the importance of motifs, transcription factors (TF) and transcription and translation process. Further, we will also summarise about some deep learning models that have seen some success having trained on active motifs. Lastly based on the background, we will present my motivation, research question and the further structure of my thesis.

2.1 Genes

The significant differences between species and within species are a result of inheritance. For example, this separates humans from chimps. A *'gene'* is a section of DNA which contains instructions for making a ribonucleic acid (RNA) molecule or a protein (Elseton & Satagopan, 2012).

Within humans, inherited genes play a role in determining skin colour, hair length and various other characteristics such as intelligence and vision. However, these differences can also be caused by environment (Clerget-Darpoux & Elseton, 2013) and *gene expressions*. Gene expression is the process by which information from a gene is used to synthesize a functional product which will allow a gene to produce protein as the final product. The Gene expression of a cell determines what the cell will do.

In 2003, the completion of the human genome sequencing by the Human Genome Project opened wide range of possibilities to study the human reference DNA sequence. The reference genomes sequence was obtained through aggregating the genome sequences of multiple people (Wheeler et al, 2008).

2.2 Deoxyribonucleic Acid (DNA)

DNA is a molecule composed of poly nucleotide chains that coil around each other to form double helix and carry a genetic structure for the growth, development, functioning and reproduction of an organism (Alberts & Johnson, 2014). The double helix structures that carry the genetic information are complementary to each other. The individual building blocks of DNA are called *nucleotides*. There are 4 types of nucleotides: *Adenine (A)*, *Cytosine (C)*, *Guanine (G)* and *Thymine (T)*. The combination of two letters is called a base pair. The whole human genome sequence obtained during the human genome project consists of 3 billion of these base pairs. The long DNA sequences with which we are made of differs with every person.

There are two kinds of DNA. *Coding DNA* and *non-coding DNA*. The mutations in coding DNA produce adequate amount of protein. The coding DNA are only 1% of the total DNA in our body. On the other hand, mutations in the non-coding DNA produces aberrant regulation of protein. The non-coding DNA sequences are the remaining 99% sequences. Many studies (Brown, 2017; Hardiman et al, 2017; Kiernan et al, 2011) have established that most terminal illness such as ALS, Alzheimer's and Parkinson's are a result of mutations in the non-coded region of the DNA. The evidence of this is that all the above-mentioned terminal illness has a common hallmark of aberrant regulation of protein (Hardiman et al, 2017; Kiernan et al, 2011).

2.2.1 Structural complexity of DNA sequences

Linguistic sequence complexity (LC) is a measure of vocabulary richness of a DNA sequence (Trifonov M, 1990; Lio et al, 2013). When a DNA sequence is written as a text using four letter nucleotides, the repetitiveness of the letters can be calculated as sequences complexity. The general idea of LC can be understood better by representing each letter in a DNA sequence with a tree structure (Figure 1). Based on this idea, most complex sequences have maximally balanced trees while the measure of imbalance serves as a complexity measure. The idea here is to understand that DNA sequences from different regions have different proportion of nucleotides and therefore the complexity associated with these sequences also differ. For example, let us consider two sequences of length 150 from blood and lower motor neuron. Both sequences would have different structure if a tree were to be drawn. Therefore, different trees have different complexities.

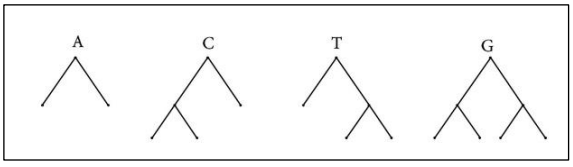


Figure 1. Trees corresponding to nucleotide bases: Combination of nucleotides in a DNA sequences is represented as a tree. The idea is that different set of DNA sequences are represented as different structured trees with a certain overall structure leading to certain protein regulation (Trifonov M, 1990).

2.3 Genetic Variants

DNA sequences are unique for every human due to genetic variants (Gibson., 2012). Genetic variants are minor changes in the string of A, C, G, T. Varying features, and characteristics among people are a result of these variants. A *variant (or mutation)* is a change to the nucleotide sequence at a particular position in the genome as compared to the human reference genome. These are also called point mutations. For example, the position of an A might be interchanged

by G, meaning it is a significant mutation. Variants can occur in a coding or non-coding regions of the DNA. Most of these mutations go unnoticed or could be dangerous or cause other disruptions within the body ultimately leading in a terminal illness.

Genetic variance is the contribution of genetic differences among individuals, with variation in *phenotype* (observable characteristics or traits of an individual). There are two types of genetic variants. First, *common genetic variants* include thousands of trait specific genes found across a large diverse population. However, common genetic variants do not explain variance across population (Wang et.al, 2019). For example, the gene associated with hair colour is *Melanocortin 1 Receptor (MC1R)*, which is a common genetic variant found across everybody. Although hair colour is determined by a common variant MC1R, the mutations within the gene in the body determine the hair colour of the person. Second, the *rare genetic variants* are relatively less in frequency and hold the view that most of the variance for certain complex diseases is due to relatively high penetrance of rare genetic variants in the non-coding region of the DNA. Higher penetrance might affect the internal molecular composition subsequently disrupting other internal functionalities and ultimately causing a disease such as ALS.

Overall, the idea is that psychiatric disorders such as schizophrenia share inherited common genetic variants that are involved in the likely precipitation of the disorder (Byrne & Heverin, 2013). There must be a substantial effect of nature and nurture for these inherited common genetic variants to manifest into a serious mental health issue. The rare genetic variants may or may not be inherited but are not observed among vast population. There is evidence that many serious illnesses such as ALS and Alzheimer's are identified through analysing in rare genetic variants in the non-coding DNA (Hardiman et al, 2017). It is likely that most ALS mutations are in fact rare genetic variant mutations occurring in non-coding DNA. Such mutations would likely act by disrupting transcription factor binding site.

2.4 From DNA to Protein: Transcription and Translation

Proteins are biomolecules which perform vast variety of functions in an organism. Proteins primarily differ from one another in sequence of amino acid which is dictated by the nucleotide sequence. The central dogma in biology is that DNA is converted into a functional product (Figure 2). This happens in two steps. First, *Transcription* is a process by which DNA is copied (or transcribed) to messenger ribonucleic acid (mRNA). mRNA is single strand molecule that carries the message of transcription. Second, *Translation* is a process in which mRNA is used to produce protein. In summary, this process of transcription and translation determines the amount of protein that needs to be produced in a coded or non-coded DNA.

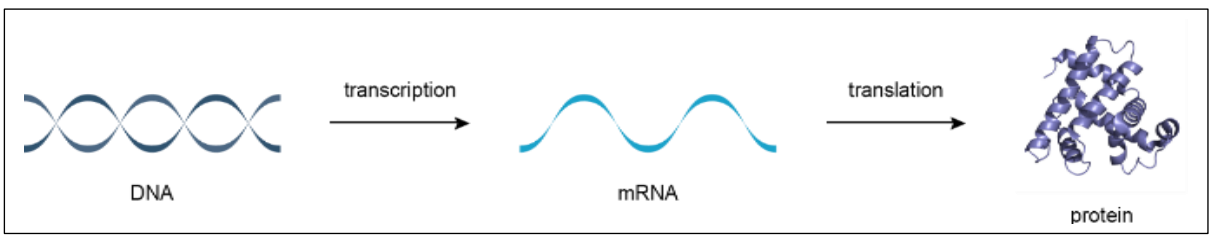


Figure 2. The central dogma in biology: The central dogma of molecular biology explains the flow of genetic information from DNA to RNA to make a functional product, a protein. The central dogma suggests that DNA contains the information needed to make all of our proteins and that RNA is a messenger (thus, mRNA) that carries this information to the protein (Leavitt, 2004).

2.5 Transcription Factor (TF)

A *Transcription factor (TF)* is a protein that controls the rate of transcription from DNA to mRNA (Latchman, 1997). The transcription process is initiated when a TF binding with the DNA. The region of the DNA to which TF binds is called *transcription factor binding site (TFBS)*. These sites are short segments of DNA that are specifically bound by one or more proteins with various functions. This binding process of proteins is stimulated by short conserved sequence elements of the DNA sequences called *motifs*.

The core purpose of a TF is to turn on or turn off the gene to make sure they are expressed in the right place at the right time in the right amount throughout the cells in the body. Groups of TFs work in coordination to direct cell division, cell growth and cell death of a living organism. TFs promote or block the recruitment of components that performs transcription of genetic information.

2.6 Motif

A sequence motif is short, conserved sub sequence element (or a nucleotide) (Colbran & Chen, 2017) in a DNA sequence that plays a key role in biological functions. The key role of a motif is to indicate sequence-specific binding sites for proteins such as TF. Other significant functions of a motif also include mRNA processing and transcription termination.

Motif in a DNA sequence permits the transcription and translation for a DNA sequence to which the motif belongs. Mutation in a motif in turn disrupts the TF to that motif, which disrupts the TFBS leading to irregular transcription leading to aberrant amount of protein being produced in the nearby. Mutation in a motif which causes disruption of transcription also alters gene expression.

ALS is a disease whose onset is attributed to such multiple combination of processes and the pathological hall mark of the disease is protein inclusion. However, the root cause for the combination of processes to start are the mutations within active motifs, on which we focus in this research project.

Understanding the complex biological mechanisms requires characterization of motifs which affects the gene expression at the transcription level. This is also one of the greatest challenges in molecular biology (Pavesi & Mauri, 2004). Transcription is modulated by the interaction of TFs with their corresponding binding sites (TFBS) mostly located near the *transcription start site (TSS)* of the gene.

2.6.1 Traditional approaches to identify motifs.

Conventional motif discovery algorithms use p*osition specific frequency matrix (PSFM)* and consensus logo. However, the consensus logo is just another way of representing binding sites and proves to be a great tool in interpreting the results in an understandable way.

Position specific frequency matrix (PSFM) are the most popular way to represent binding sites (Figure 4). These are matrices provide information on the frequency of each base in each position of the DNA binding motif (Schneider & Stormo, 1986). PSFM has an implicit assumption that different positions at the DNA binding site contributes independently to the site function. Ultimately, these PSFM result in a visualization known as the ‘consensus

sequence logo' derived from consensus sequence that can further be easily interpreted. A consensus sequence is a calculated order of most frequent residual nucleotides that are found at each sequence (Figure 3). It represents the results of multiple sequence alignment in which related sequences are compared to each other and similar sequence motifs are subsequently calculated.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	1	0	1	5	32	5	35	23	34	14	43	13	34	4	52	3
C	50	1	0	1	5	6	0	4	4	13	3	8	17	51	2	0
G	0	0	54	15	5	5	12	2	7	1	1	3	1	0	1	52
T	5	55	1	35	14	40	9	27	11	28	9	32	4	1	1	1
Sum	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56

Figure 3. Position specific frequency matrix (PSFM): The top row represents the number of DNA sequence, and each with length 56. The Colum comprising of A, C, G, T represent the frequency of each nucleotide in each sequence (Wang & Xu, 2016).

DNA binding sites are such sites within DNA sequence where other molecules may bind. DNA binding sites are often associated with transcription factors and are this linked to transcription regulation. DNA binding sites can be referred as short DNA sequences that are specifically bound by one or more protein molecules. A collection of DNA binding sites can also be referred to as DNA binding motif. DNA binding motifs can be represented by a consensus sequence (Schneider., 2002).

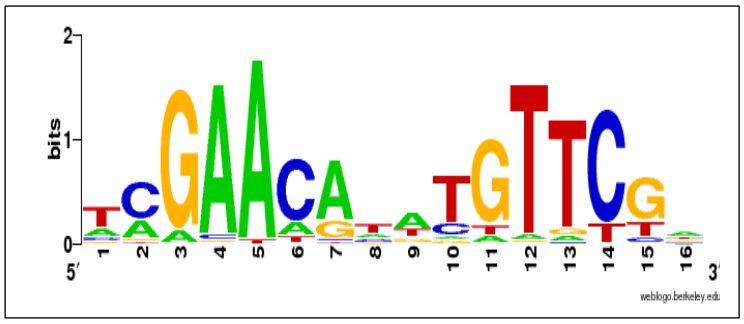


Figure 4. Example consensus logo: A consensus sequence logo is an alternate representation of PSFM. The numbers on the x-axis represent teach individual sequence of a specific length (Wang & Xu, 2016). The bits on the y-axis describes the importance of nucleotide in a given sequence. For example, the nucleotides in the third sequence is represented as 'GA'. In that specific sequence, there is always a G instead of any other nucleotides. However, if the sequence demands a change (or mutates), it can only change with an A in that sequence.

Existing computational approaches such as consensus sequence or position weight matrix were used for observing TFBS. But these methodologies are not generally used to predict TFBS or to classify sequences using binding sites (or motifs) (Yue & Wang, 2018). In bioinformatics one can distinguish between two separate problems regarding DNA binding motifs. One, searching for additional members of a known motifs (site search problem). Two, discovering novel DNA binding motifs in collection of functionally related sequence (sequence motif discovery; Eril & O'Niel, 2009). Many of the methods used rely on PSFM and consensus sequence. However, few authors used the deep learning approaches. Deep learning is appropriate for sequence motif discovery because it assumes that a set of sequences share a binding motif for functional reasons.

2.6.2 Motif Mining using Deep learning.

In this section, we briefly summarise the existing deep learning models that have been trained on active motifs to check the reliability to identify novel motifs.

In recent years, deep learning has achieved great success in various application domains in identifying spatial features. This makes researchers attempt to identify motifs using deep learning methodologies. Deep learning approaches have outperformed other methodologies in predicting gene expressions (Yue & Wang, 2018; Beer & Tavazoie, 2004).

There are three main types of deep learning frameworks that are used in motif mining. However, we will only discuss two main types that have proven to be successful in novel motif identification and the ones that I use in this project. These are Convolutional Neural network (CNN) and a Hybrid (CNN + LSTM) model which is a combination of CNN with long-short term memory model (LSTM).

In 2015, DeepBind (Alipanahi et al, 2015) is the first attempt to identity DNA binding sites (motifs), having been trained on balanced dataset high throughput raw DNA sequences comprising of motifs as active and inactive reference sequnces using a CNN. The aim of the DeepBind model was to identify damaging mutations in the rare genetic variants. The sequences have varying length (14-101 nucleotides) and a binding score are binary class labels. The models are trained on a balanced dataset comprising of two classes. The sequences with a class label 1 comprise of a TFBS and the sequences with a class label 0 do not have a TFBS. The reference sequences extracted specifically belong to a localised region. The DeepBind computes the probability if sequences of varying length has a TFBS. In the presence of a TFBS, the model predicts the class as 1. The models were evaluated on its ability to characterize DNA-binding protein. This model has three layers. A convolutional layer, a pooling layer, and a fully connected layer. This model acted like a precedent for deep learning in motif mining and provides a basic framework for subsequent models.

In 2017, DeepSNR (Salekin & Zhang, 2017) model was trained on a balanced dataset with DNA sequences of length 100 which are known to contain TFBS. The sequences comprising of TFBS are labelled as 1 and random DNA sequences with no TFBS as 0. The final output of the network is a binary label and the associated probability of whether a given sequences comprises of a TFBS (label = 1) or not (label 0). In other words, the model identifies a single nucleotide in each sequence that could be a TFBS. The sequences used were fixed length of 100. The DeepSNR model was able to identify transcription factor binding location at the level of a single nucleotide. With the basic structure of DeepBind, DeepSNR includes a deconvolution network which reduces the size of the activation. The model performs with a precision and recall of 87% and 77%, significantly outperforming motif search-based algorithm until then.

2016 was the first time when DanQ (Quang & Xie, 201), a CNN + RNN model was implemented in motif mining. The DanQ model attempts to identify presence of a motif that affect the regions around a gene. Each input sequence is of length 1000 nucleotides. Although the model uses the same architecture from DeepBind, it does not adopt training the model on variable length sequences. It uses fixed length sequences of size 1000 and a balanced dataset with the sequences comprising of motifs labelled as 1 and sequences with no motifs are labelled as 0. This model uses a CNN with bidirectional LSTM. The first layer of the DanQ aimed to scan the position of the motif using a convolutional layer, max pooling layer followed by LSTM layer. The advantage of this combined model was that it could identify and capture the long-term dependencies between sequence features by learning the features extracted from the convolutional layer. The training set of the DanQ model is same as the DeepBind model. The test set of the DanQ model is the same as the DeepBind model. Statistically comparing both

model predictions return a high correlation. The DanQ model achieved around 50% in the area under precision-recall curve.

In 2017, BiRen (Yang & Liu, 2017) built a hybrid architecture to predict promotor regions using only DNA sequences. The model tries to learn common promotor patterns based on the structure of the DNA across species to predict promoter sequences in humans and mouse as a binary label. The BiRen model uses high throughput sequences. The high throughout sequences enables the TF bind with the sequences. It was observed that hybrid model's bidirectional RNN in this case performs better on longer sequences. The model demonstrates excellent accuracy of 95% in predicting promoters in the humans and mouse DNA sequences.

***

3. Research Question

Having established the successful performance of the deep learning models in identifying mutations, we set up the research questions.

RQ 1: Determine the efficiency with which active motifs identify active sequences that comprise of damage causing mutations.

RQ 2: Are the prediction probability of the damage causing mutations as identified by the deep learning model directly proportional to the known effect size of mutations as determined by the standard expression quantitative trait loci (eQTL) framework.

To benchmark the model, I use Genotype tissue expression (GTEx) and Project MinE datasets. GTEx data consists of mutations that are known to disrupt transcription at the mRNA level. These mutations are identified using expression quantitative trait loci (eQTL) framework which is used to identify the effect of a mutation at the mRNA level. Similarly, we also test our model on the Project MinE that is a genome sequencing data set which consists of all mutations as observed in ALS patients and controls regardless of whether they disrupt TSBS or other functional motifs. Finally, I compare the model predicted probability with scores that represent the known effect of such mutations.

4. Structure of thesis

In the upcoming sections, we will first conduct the preliminary study (Section 5). We will elaborately discuss the process of preparing the dataset (Section 5.2) that is used to classify active and inactive reference sequences based on the contents of nucleotide using a linear regression model (Section 5.3). We also implement two machine learning models (Section 5.4) to evaluate and compare the performance of machine learning and linear regression model.

Next, in Section 6 I discuss the need for a deep learning model (Section 6.1). In Section 6.2 'Models trained on blood' we discuss the efficiency of deep learning models on blood lymphoblastoid. Lymphoblastoid are white blood cells. Mutations in lymphoblastoid are known to effect transcription. In Section 6.3, we train the models on a more efficient information. We train the models on transcription start site (TSS) in lymphoblastoid. TSS regions are where the process of transcription is initiated. Therefore, mutations in the TSS regions have the strongest effect and thus should provide robust information for classification. We compare the performance of the model in section 6.2 and 6.3. Having determined the efficiency of the model in identifying mutations that effect gene expression, we train my model on non-coding sequences from the lower motor neuron to predict mutations that cause ALS.

In the Result (Section 7), we compare all the models and infer which of the proposed models is better predicting motifs. In Discussion (Section 8), we compare the CNN and hybrid model to evaluate their model performance. Towards the end in Further Work (Section 9) we put forward a novel expectation pooling framework and other improvements that could be made in my model architecture that can be used to improve my model performance and then we conclude (Section 10) with the final take away points.

***

5. Preliminary Study

5.1 Introduction

In this section, we will first discuss about preparing the dataset to conduct our preliminary study. Next, we will implement a logistic regression model and attempt to classify DNA sequences based on the proportion of nucleotides. Finally, we will implement a naïve machine learning (ML) model to check its reliability in classifying active and inactive DNA sequences based on the content of nucleotide.

5.2 Dataset.

A typical dataset in bioinformatics comprises of a chromosome number *(chrom)*, chromosome start position *(chromStart)*, chromosome end position *(chromEnd)* and the corresponding strand information (positive or negative). In our case, this information comprises non-coding DNA sequences of active regions in the lymphoblastoid (Figure 5). Lymphoblastoid are immature white blood cells in the spinal cord. The R package *GenomicRanges* (Lawrence et al, 2013) helps us extract the fixed size DNA sequences using the chromosome start and end position. This gives us our active sequences. Therefore, we label these active sequences as 1.

1	9976	10725	id.1	1000	+
1	235351	235950	id.2	1000	+
1	564976	570525	id.3	1000	+
1	713701	714750	id.4	1000	+
1	762751	763200	id.5	1000	+
1	804976	805650	id.6	1000	+
1	825001	825300	id.7	1000	+
1	839851	840375	id.8	1000	+
1	840451	841050	id.9	1000	+
1	859051	859350	id.10	1000	+

Figure 5. The raw file used to extract active reference sequences: Each row consists of a chromosome start and end position which comprise sequences of width 1000. Each of the selected regions comprise of active motifs in non-coding DNA sequences of the lymphoblastoid.

A similar approach is used to extract inactive sequences. In my project, the inactive sequences are extracted from the human reference genome (Homo_sapiens.GRCh37.75.gtf.gz; ftp://ftp.ensembl.org/pub/release-75/gtf/homo_sapiens/Homo_sapiens.GRCh37.75.gtf.gz). The human reference genome consists a record of sequences as observed in an average human. The inactive sequences are randomly extracted fixed length DNA sequences from the human reference genome. Thus, I label these inactive sequences as 0. Thus, we have a balanced dataset with a total of 500 active and inactive reference sequences (Figure 6).

Having extracted out DNA sequences, we check the average proportion of nucleotides A, C, G and T in both the active and inactive reference sequences. In Table 1, we see that the proportion of CG content is higher in active sequences as compared to the inactive sequences. The higher CG content is a characteristic generally observed when sequences are extracted from the active regions. However, we expect the binary machine learning model to classify active and inactive DNA sequences based on this bias in the CG content.

Sequences	Label
TGGTT. . . GCATT	1
CCCAC. . . ACTTT	1
CCACT. . . GCCAA	1
GGTGC. . . CTTTC	0
ACAAG. . . CCTGC	0
GAGGT. . .CAGGG	0

Figure 6. A sample representation of the training set: The training set comprises of active and inactive reference sequences extracted from the human reference genome. The active sequences are represented as 1 and some of the active sequences comprise of active motifs.

We further also compare the root mean square error (RMSE) to check the discrepancy in the data. RMSE is used to find the difference between the actual and predicted value. In our case we assume that the actual values are positive reference sequences and the predicted value as inactive reference sequences. We find that the base composition of RMSE measured between the active and inactive sequences is 0.0727. This means that bias among the active and inactive DNA sequences in very high.

Class	*A*	*C*	*G*	*T*
Active - Lymphoblastoid	0.2266	0.3466	0.3066	0.1200
Inactive – Human reference genome	0.2715	0.1945	0.1950	0.2725

Table 1: Proportion of nucleotides in active and inactive reference sequences: An imbalance in the proportion of nucleotides is seen in active reference sequences. This CG bias is mostly seen in active regions.

5.3 Machine Learning

5.3.1 Encoding sequences

Before we directly feed the raw DNA sequences to the machine learning models, we first convert these sequences using encoding to make the data understandable for the model. An efficient machine learning models prefers its input variable to be numeric matrix. Therefore, the DNA sequences need to be transformed to convert these nucleotides into a matrix.

We choose one-hot encoding which is a representation of categorical variables as a matrix of numerical values. Each DNA sequence consists a combination of nucleotides represented as A, C, T and G. Each nucleotide is encoded as a vector of binary values [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0] and [0, 0, 0, 1]. The encoded nucleotides are then stacked to create a matrix (Zhou et al., 2019). Thus, each sequence is represented as a combination of these encoded nucleotides. For example, a sequence ‘AATGC’ would be encoded as [[1, 0, 0, 0] [0, 1, 0, 0] [0, 0, 1, 0] [0, 0, 0, 1] [0, 1, 0, 0]]. Like encoding sequences, we one hot encode the labels 0 as [1, 0] and 1 as [0, 1]. that differentiate our active and inactive reference sequences. Therefore, all the training input sequences are matrix of binary vectors.

5.3.2 Model Architectures

Deep learning is a subset of machine learning that comprise of multi-layered deep neural networks and rely on large complex datasets. A deep learning model trains on a large dataset to find a structure in the given data which is used to make the prediction by the models. The models are then validated on known results to check the efficiency of the model. Deep learning in motif mining has seen much success in identifying novel motifs as discussed in section (Section 2.6.2).

In this section, we build two (CNN, CNN+LSTM) deep learning model architectures and train them on active sequences from the lymphoblastoid and inactive sequences which are randomly sampled from across the human reference genome. The successful performance of the model is established when the model can classify active sequence (labelled as 1) from the inactive sequence (labelled as 0).

Convolutional Neural Network (CNN)

The proposed CNN to identify sequences comprising motif is a sequential model that comprises of two one-dimensional convolutional, one max-pooling layer followed by two dense layers (Figure 7). The CNN layer acts as feature extractor that transforms input DNA sequence into multidimensional feature representation. Each CNN layer comprises of 32 filters with kernel size of 64 and input dimension of (150, 4). The max pool layer of size 5 aggregates the extracted features from the convolutional layers. The aggregated features are flattened and then passed to the dense layers which will perform matrix multiplication resulting in weights that can be trained and updated.

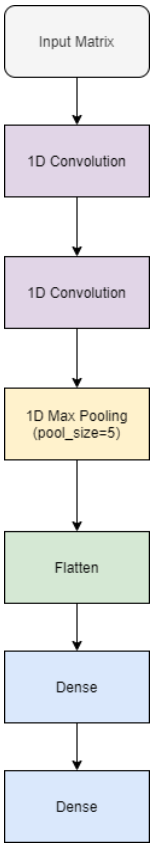


Figure 7. Architecture of convolutional neural network: Block diagram representing the architecture of the CNN.

Hybrid (CNN + LSTM) Model

Like CNN, the hybrid model also inputs one hot encoded sequences of length 150 bp and consisting of all nucleotides. The CNN layer acts as feature extractor that transforms input DNA sequence into multidimensional feature representation (Figure 8). Each CNN layer comprises of 32 filters with kernel size of 64 and input dimension of (150, 4). The max pool layer with pool size 5 aggregates the extracted features from the convolutional layers. The aggregated features are passed to layer with 50 units, a recurrent drop out of 1% and an activation function of rectified linear unit (ReLU). ReLU returns 0 if it receives any negative input and a positive value greater than 0 for any positive value. On longer sequences, ReLU works better because of its linear learning as a part of its function. The derived features are further passed to dense layers which return a predicted class label (0= inactive reference sequence/ no motif, 1 = active reference sequence/ presence of motif)

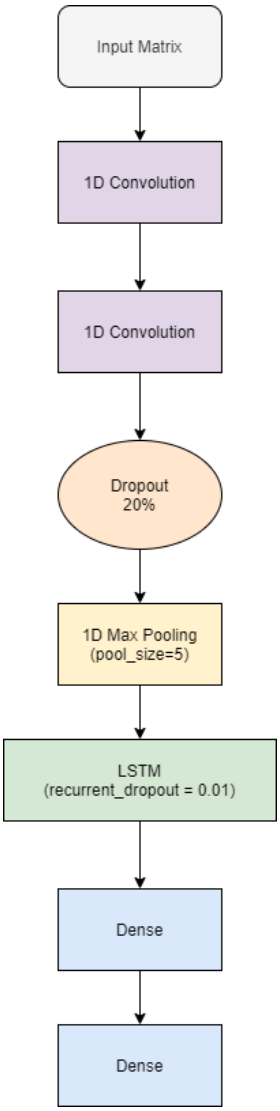


Figure 8. Architecture of the hybrid (CNN + LSTM) model: Block diagram representing architecture of the hybrid (CNN + LSTM) model.

Model Compilation

We compile the model with the loss function of *binary cross entropy* and the optimizer '*adam*'. The adaptive movement estimation (adam) computes the adaptive learning rates for each parameter. Binary cross entropy is the measure of difference between two probability distributions for predicting class 1. A cross entropy value of 0 is a perfect score.

5.3.3 Training model.

Except the differing architecture of the models, the other steps involved remain the same for both models. With the balanced dataset of size 500. We now split the data into 70% and 30% to train and test out models. We train my model on the 70% of the data for 50 epochs with a validation split of 10% to include validation set. The validation set is used to select the best performing algorithm. Whereas the 30% test set is used to estimate the generalization error of these models. Finally, we use a confusion matrix to assess the model performance.

5.3.4 Model Performance

In the Figure 9 (A) and (B), we see that both the models have a high accuracy in classifying sequences with active motifs in lymphoblastoid from the inactive human reference sequences. The accuracy of the CNN model on the test set is 92.44% with validation loss of 0.055. Validation loss returns a value that determines how poorly or well a model behaves after each iteration. Similarly, the accuracy of Hybrid model on test set is 99.16 and a test set loss of 0.02.

Comparing the loss of both models, we can infer that both the models show a learning curve. We see that the training and validation loss are like each other with the training loss greater than test set loss with reasonable oscillation. The higher accuracy may be because of the bias in the CG content which is a characteristic of sequences extracted from active regions.

We evaluate the performance of the models using a confusion matrix to derive the following performance metrics: error rate, sensitivity, precision, F1, Mathew correlation coefficient. The *error rate* is calculated as the number of all incorrect predictions divided by the total number of datasets. The closer the value is to 0, the better the model performance. *Sensitivity* is also known as true positive rate which is defined as the number of correct positive predictions divided by the total number of positives in the dataset. *Precision* is defined as the number of true positives divided by the number of true positives plus the number of false positive. *F1 score* is the harmonic mean of precision and recall.

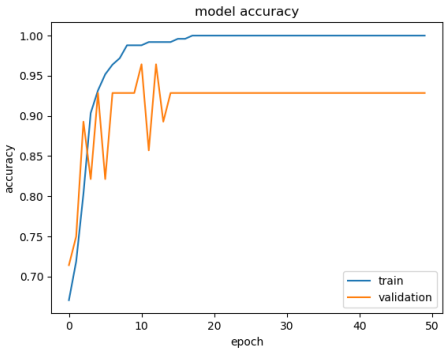
$$Error\ Rate = \frac{\text{False Positive } + \text{ False Negative}}{\text{True } + \text{ False}}$$

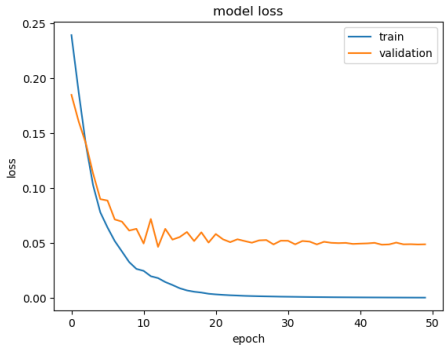
$$Sensitivity = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Negative}}$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Positive}}$$

$$F1\ Score = \frac{2\ (\text{Precision})\ (\text{Recall})}{\text{Precision } + \text{ Recall}}$$

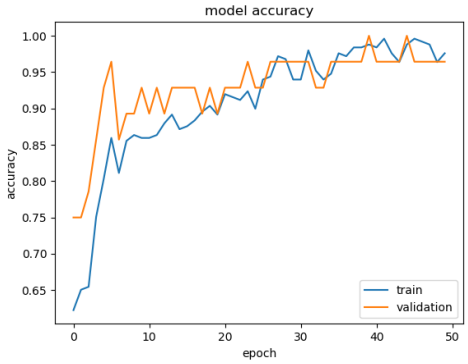
(A)

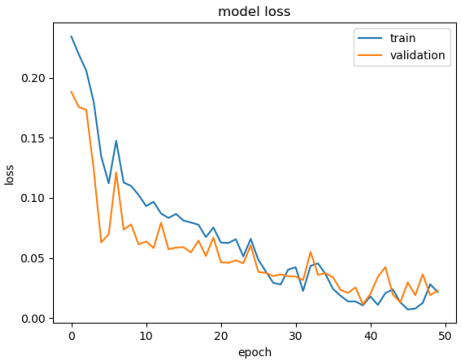




	ACTUAL		
		0	1
PREDICTED	0	32	4
	1	6	77

(B)





	ACTUAL		
		0	1
PREDICTED	0	32	4
	1	2	81

Figure 9. Performance of CNN (A) and hybrid (B): Boxes (A) and (B) represent the performance of the CNN and hybrid model. Both the models perform efficiently in classifying active and inactive reference sequences based on the contents of nucleotide. Each box shows the model accuracy, model loss and the resulting confusion matrix corresponding to each of CNN and hybrid.

$$Specificity = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Negative}}$$

$$Mathew\ Correlation\ Coefficient = \frac{(\text{TP}.\text{TN}) - (\text{FP}.\text{FN})}{\sqrt{(\text{TP} + \text{FP})\ (\text{TP} + \text{FN})\ (\text{TN} + \text{FN})(\text{TN} + \text{FN})}}$$

The CNN model performs with test error rate of 0.08. The sensitivity of the model is 0.395 and the precision is 0.888. The error rate of the Hybrid model is 0.05. The sensitivity of the model 0.94 and a precision of 0.888. These metrics show that the deep learning models are efficient in classifying active and inactive reference sequences based on the CG content in the proportion of nucleotides.

5.4 Summary

In summary, we attempt to classify the active and inactive reference DNA sequences based on the proportion of nucleotides. We implement neural networks that could classify such sequences. We see a high-performance accuracy by both the CNN and hybrid. This is because of the characteristic associated active DNA sequences extracted from active regions, the CG bias. Therefore, any simple deep learning model that attains a high accuracy is because of the biased CG content within the active sequences. Thus, to build a more efficient deep learning model, we must overcome the CG bias. To do so, we normalize the contents in both active and inactive DNA sequences such that the proportion of the nucleotides remain consistent. This is a significant step so that we build a model that rather learns the complexity associated with the active regions rather than classifying based on disproportional content of nucleotides. Therefore, the resulting dataset after normalization should consist of same proportion of nucleotides in both the active and inactive reference sequences.

6. Deep Learning

6.1 Introduction

In this section, we build my two (CNN, CNN+LSTM) deep learning model architectures and train them on the sequences intrinsic to three different regions (a) Lymphoblastoid, (b) Transcription start site and (c) Lower motor neuron. To validate and benchmark my model, I test the models on GTEx dataset that comprise of the significant mutations as identified by the standard eQTL framework. Similarly, we identify the efficiency of these models to predict ALS mutations from the known true positives in the Project MinE dataset.

This section is divided into 2 parts: (a) Blood Data (b) lower motor neuron. In the first part, we train our model architectures on the data related to blood. We train our models on DNA sequences from the blood lymphoblastoid and regions around transcription start site within the lymphoblastoid. We then validate our models on the GTEx and MinE dataset to determine the efficiency of the model in identifying mutations. In the second part however, we train my model on DNA sequences from an organ, the lower motor neuron. Similarly, we validate the performance of the lower motor neuron model only on the MinE validation dataset.

To test the performance of the models trained on blood I use two datasets. First, expression quantitative trait loci (eQTL) based Genotype-tissue expression (GTEx; Londsale et al, 2013) project. It is a tissue database collected across 1000 individuals to study the relationship

between genetic variation and gene expression in human tissue (Lonsdale et al, 2013). The eQTL framework offers a powerful approach to explain the genetic components underlying altered gene expression. Studies primarily in blood, liver, skin, and brain indicate that eQTLs are common in humans.

Second, we use a database collected from ALS patients. Project MinE (Project MinE ALS Sequencing consortium, 2018) is a database consisting of single nucleotide motifs as observed in ALS patients. Most of the mutations in MinE (Project MinE ALS Sequencing consortium., 2018) do not cause ALS and a small number result in damaging motif disruptions. This database was created to understand the genetic basis of ALS. MinE database aims to analyse the DNA of at least 15,000 ALS patients and 7,500 controls.

To benchmark the models, we compare the size of the mutation as predicted by the models with Project MinE. We use *phred scores* from the *Combined Annotation -Dependent Depletion (CADD;* Rentzsch et al, 2019*).* CADD scores rank single nucleotide mutations throughout the human reference genome. Phred scores range from 0 to 50. Phred scores are normalised scores which are based on the genome-wide distribution of scores given to ALS mutation across species. For example, a phred score of 10 or greater indicates that such a mutation in the top 10% of all possible mutations reference genome. A score of greater than 20 indicates that such a mutation is in the top 1% of all possible mutations in the reference genome.

Similarly, to compare my model performance on GTEx, we use the *effect size* derived using the standard eQTL framework. Effect size is defined as the slope of the linear regression and is computed as the effect of the mutating Alt nucleotide relative to Ref nucleotide in the human reference genome. Effect size is computed in a normalised space where magnitude has no direct biological interpretation. Effect size ranges from -3 to 3. The mutations having an effect size greater than -1 and 1 are supposed to have stronger effect.

6.2 Part 1: Models on blood lymphoblastoid

The upcoming deep learning models in Part 1 (Section 6.2) and (Section 6.3) train on non-coding DNA sequences from white blood cells (lymphoblastoid) and on transcription start site region (where mutations have significant effect) in white blood cells. We train the models on large dataset comprising of active DNA sequences from active regions (such as lymphoblastoid, TSS and lower motor neuron) and inactive reference sequences extracted from the human reference genome to check the validity of the model in predicting mutation that disrupt transcription thereby altering gene expression using GTEx and its validity in predicting mutations that also disrupt transcription leading to disruption of TFBS causing ALS using Project MinE. We expect to see that the models trained on blood should be better able to classify GTEx mutations than MinE. The objective of the training set for the model is to learn what it takes to classify an active sequence from an inactive sequence. Having learnt the differences, we check the model if its learning is sufficient to identify active GTEx sequences as observed real time.

The idea of the training set in our case is that not all active reference sequences have motifs within them. Some of the active sequences might have motifs and some might not. However, the regions used to extract the training active reference sequences are known to comprise of motifs. The general idea is that we are trying to implement a model which learns the presence of a motif among long sequences (which is usually the case in our bodies). Having learnt the presence of a motif based on the structure of sequences, how efficiently can our models identify

the real motifs in GTEx and MinE. The objective of the upcoming models is to learn the motifs in non-coding DNA sequences in blood. Having learnt the motifs, the efficiency of the models will be established on their ability to identify active sequences from GTEx and MinE. We expect to see that the models identify more GTEx mutations that MinE mutations. This is because, the mutations in blood rather disrupt the process of transcription than break the TF and their TFBS. The idea hear for the model is to learn the amount of activation it takes to classify active sequences from the inactive sequences in blood. Having learnt the activation required, we establish the model efficiency in identifying active sequences from GTEx and MinE. However, towards the end of this section we see that the GTEx and MinE sequences show some subtle effect due to which my models fail to identify active sequences thereby giving us a low model performance on the test set.

6.2.1 Training Set

In Section 2.6.2 we see that the models have high classification accuracy when trained on high throughput active reference sequences and random inactive reference sequences from the human reference genome. Therefore, the training set that we use also comprises of active and inactive reference sequences. The active sequences are high throughput Assay for Transposase-Accessible Chromatin (ATAC-sez; Buenrostro et al, 2015) sequences from active lymphoblastoid. There is a difference between active and inactive reference sequences. The active ATAC sequences are loosely bound by DNA. On the other hand, the inactive reference sequences are tightly bound by the DNA. The loosely bound active reference sequences let large number of TFs bind with DNA. This also means that more motif mutations can be tested. On the contrary, the inactive reference sequences allow fewer TFs to bind with DNA and hence fewer TFs can bind with DNA allowing fewer motif mutations. In this project we use active ATAC sequences corresponding to the active regions (lymphoblastoid, TSS, lower motor neuron) and the inactive reference sequences randomly sampled from the human reference genome. The active sequences comprise of motifs and the inactive sequences comprise of no motifs. The active sequences are labelled 1 and the inactive sequences are labelled 0.

In the section (Section 5.2; Figure 5) we saw a raw file that comprises of regions associated to lymphoblastoid. We use a full genome sequencing package such as B*SGenome.Hsapiens.UCSC.hg19* (Patel & Gorham, 2018) to extract DNA sequences of length 150. We base the choice of sequence length from section 2.6.3. The successful networks that identify motif mutations train their model on sequences of length less than 150. To do so, I use an inbuilt function *getSeq()* from the *GenomicRanges* (Lawrence et al, 2013) package. The getSeq() function extracts the DNA sequence along with the proportion of nucleotide from their corresponding start and end position from the raw lymphoblastoid file. This gives us our non-coded active DNA sequences. Similarly, we randomly sample inactive reference sequences from the human genome ensemble and extract one inactive reference sequence for each active reference sequence from the aggregated human reference genome ensemble along with their proportion of nucleotides. Doing so gives us our inactive sequences.

Normalising Training Set

I find the mean squared difference among the same nucleotide frequency of the active and inactive DNA sequences and subsequently calculate the root of mean square difference. For example, I find the mean square difference between the number of nucleotide (A, C, G, T) from a single active and inactive reference sequence (A_{active}-$A_{inactive}$, C_{active}-$C_{inactive}$, G_{active}-$G_{inactive}$, T_{active}-$T_{inactive}$). We then find the mean across these differences (error = mean ($A_{diff,}$ C_{diff}, G_{diff},

T_{diff}) * 2). This will result mean square error. Finally, calculating the root of the mean difference (√error) should give us a value that should be less than 0.02. This is an iterative process until the RMSE among all pairs of active and inactive DNA sequences is less than 0.02.

Having such low RMSE ensures that there is no bias among the proportion of nucleotides for a pair of active and inactive reference sequence. This means that the number of A, C, G and T in both the active and inactive sequences will be the same. This process makes sure that the proportion of nucleotides always stay consistent in active and inactive sequence. Therefore, we have a balanced dataset in which the RMSE between each pair of active and inactive sequences is less than 0.02. This process remains the same for generating sequences for the data set in upcoming sections (Section 6.3 and Section 6.4).

As a step for pre-processing data, some of the inactive reference sequences comprise of 'NNNNN…' and these are mainly seen in the human reference genome from which we extract normalized inactive reference sequences. In the human reference genome 'NNNNN…' represents that it is not yet known which combination of nucleotides occur in the area. To always maintain a balanced dataset, we remove the inactive reference sequences comprising of NNNNN...'s and their corresponding active reference sequence. Therefore, to train my models on lymphoblastoid, we have a balanced training set of size 140,865 active and inactive reference sequences extracted from the lymphoblastoid. These sequences are further one-hot-encoded as in Section 5.4.1 to train the model.

6.2.2 Test Set

A raw file with motifs.

A test set with active motifs that destroy the TFBS looks as in Figure 10. A text file with active motifs comprises of the chromosome (Chrom) to which the motif belongs, the start position (chromStart) and the end position (chromEnd) of the motif. It also consists of a reference (Ref) and alternative (Alt) column. The Ref column shows actual motif as generally observed. The Alt column shows the mutation leads to destruction of TFBS.

The general concept of testing the model is as follow. For example, in the first row of the Figure (10. (B)), we see that the actual motif 'G' leads to a normal transcription and translation process. Other mutations such as 'C' or 'T' might not be destructive. However, if the mutations cause 'G' to be replaced by 'A' such a mutation is known disrupt the TF of 'G' which disrupts the TFBS leading to irregular transcription thereby causing aberrant protein regulation in the nearby region leading to the onset of ALS.

(A) GTEx

Chrom	chromStart	chromEnd	Ref	Alt	Width	Strand
chr1	33077643	33077644	C	T	1000	+
chr1	33077643	33077644	C	T	1000	+
chr1	33077686	33077687	C	T	1000	+
chr1	33077686	33077687	C	T	1000	+
chr1	33077690	33077691	C	A	1000	+
chr1	33077690	33077691	C	A	1000	+
chr1	33077716	33077717	A	C	1000	+
chr1	33077716	33077717	A	C	1000	+
chr1	33077730	33077731	C	G	1000	+
chr1	33077730	33077731	C	G	1000	+

(B) Project MinE

chr1	894519	894520	G	A	b37
chr1	894573	894574	G	A	b37
chr1	894719	894720	C	T	b37
chr1	894890	894891	A	AAGAC	b37
chr1	895037	895038	A	AG	b37
chr1	895706	895707	G	A	b37
chr1	895820	895821	G	C	b37
chr1	895889	895890	A	T	b37
chr1	896064	896065	C	G	b37

Figure 10. The raw file comprising of active motifs in (A) GTEx and (B) Project MinE: I use these files to generate the test set on which I evaluate the performance of the models trained on blood cells.

To check this concept, we need to generate the reference and altered sequences from their designated chromosome positions using the motifs from both files GTEx and MinE. Once we train and proceed to test the efficiency of our model, we predict the labels and their corresponding probabilities for both the reference and the altered sequences. To determine the efficiency of the model we look for reference sequence with a prediction 1 and their altered (or mutated) sequence with a prediction of 0. Subsequently, finding the absolute difference between the probability of prediction determines the size of mutation or the difference in change as predicted by the deep learning models (size of mutation = $[P_{reference} - P_{altered}]$). *We hypothesize if the mutations with high effect are directly proportional to the size of mutation*. The mutations which show such direct proportionality could be disease causing motif mutations.

Generate reference and altered test sequences.

To generate the reference and altered test sequences, we only consider single nucleotide mutation. First, we delete the rows from both Ref and Alt with more than 1 nucleotide. The idea is to first generate reference sequences and then replace the 75^{th} nucleotide (one in the centre) with the nucleotide in Alt (Figure 10).

We use the BSGenome.Hsapiens.UCSC.hg19 full genome sequencing package because it provides inbuilt functions for DNA sequence manipulation and generating sequences. We use *start()* and *end()* to generate DNA sequences using the reference nucleotide and their corresponding start and end position. Performing *start(Ref) – 74' and 'end(Ref) + 74* adds nucleotides from human reference genome on either side of the reference nucleotide. This gives us our active reference sequences. Subsequently from the same package we use the function *replaceAt(pos, IRanges(75,75), Alt.list)* to replace the 75^{th} nucleotide in the reference sequence with the nucleotide in Alt.list. Doing this gives us our altered or mutated sequences (Figure 11).

The objective of the models is to predict the associated class (0= inactive/ 1= active) and the probability of the class prediction for both the reference and altered sequences. Since the reference sequences comprise of true positives, we look for such sequences in GTEx and MinE whose reference class label is predicted as 1 and the altered class label is predicted 0. This will lead us to the possible mutating sequences. Subsequently, we find the absolute difference between the probabilities of the associated class prediction for reference and altered sequences like mentioned in the above section. The resulting difference between the probabilities describes the size of mutation (size of mutation = $[P_{reference} - P_{altered}]$). Finally, we compare this size of mutation with the effect size from GTEx and phred score from MinE to benchmark the model predictions. We follow the same steps for both the GTEx and MinE data. This gives us a total of 17,879 GTEx sequences and 16,788 project MinE sequences. Thus, the total size of the test set comprising of GTEx and MinE data is 34,667. The test set remains the same for the models trained on blood data.

Comparing the model performance with existing scores

To derive the scores from the same regions that we derive the test sequences, we use the mathematical *intersect* function. We intersect the raw file comprising of motifs with the text file comprising of scores, using *bedtools intersect* (Quinlan & Hall, 2010). *Bedtools* is a package in Linux to perform terminal command operations involving DNA sequences. Using bedtools intersect gives the scores corresponding to the chromosome positions which is used

derive the test set comprising of reference and altered sequences. Therefore, I have complete test set which involves 34,667 samples involving 17,879 GTEx and 16,788 MinE sequences comprising of known active motifs along with their corresponding effect size (for GTEx) and phred score (for MinE). The test set remains the same for both the models trained on blood.

(A)

Ref	Alt
G	A
G	A
C	T
G	A
G	C

(B)

Reference Sequences:

"CCGCTGAGAGG" "AAATCGGCCCT" "CGCCGCGCCAC" "GAGGAGGATGG" "GGATGGCGGGG"

Mutated Sequences:

"CCGCTAAGAGG" "AAATCAGCCCT" "CGCCGTGCCAC" "GAGGAAGATGG" "GGATGCCGGGG"

Figure 11. Shows the substring of sequence between 70 – 80 from a total size of 150: The above box (A) represents the reference (Ref) and altered nucleotide (Alt). Box (B) represents the reference sub sequence and its mutated sequences.

6.2.3 Model Architectures

The average performance of the model is determined using a training set to establish efficiency with which the models classify active sequences from the active regions (lymphoblastoid, TSS, lower motor neuron) from the inactive human reference sequences. The successful performance of the model is evaluated on the test set in which the model can classify active GTEx sequence (labelled as 1) from the active MinE sequence (labelled as 0).

Convolutional neural network

The proposed CNN is a sequential model that comprises of two one-dimensional convolutional layer that accepts a single encoded sequence of length 150 and a dimension of 4 (Figure 7). The CNN layer acts as feature extractor that transforms input DNA sequence into multidimensional feature representation. Each CNN layer comprises of 32 filters with kernel size of 64 and input dimension of (150, 4). The max pool layer of size 5 aggregates the extracted features from the convolutional layers (Luo & Tu, 2020). The max pooling layer picks a single feature in each patch of a single encoded sequence. The aggregated features flattened and then passed to the dense layers with which perform matrix multiplication resulting in weights that can be trained and updated.

We use 5-fold cross validation to evaluate the performance of the model on the complete training set. It provides a robust estimate of the performance of a model on unseen data. The cross-validation algorithm randomly splits the training data in 5 subsets and takes turns to train the model on all subsets except the one which is held out. This process repeats until all subsets are given an opportunity to be a held-out validation set. The performance measure is averaged across all models that are created. The models are compiled the same way as in section 5.4.2.

I train the CNN model architecture on 140,865 active and inactive reference sequences. The RMSE between the active and inactive reference sequences is less than 2%. The average accuracy across the 5 folds for the CNN model on the training set is 69.27% with a high loss of 0.63. However, the model performs consistently in every fold (Figure 12. (A)).

Hybrid (CNN + LSTM) Model

Like the CNN, the hybrid model also inputs encoded sequences of length 150 and consisting of all nucleotides (Figure 8). The two CNN layers act as feature extractor that transforms input DNA sequence into multidimensional feature representation. Each CNN layer comprises of 32 filters with kernel size of 64 and input dimension of (150, 4). The max pool layer of size 5 aggregates the extracted features from the convolutional layers (Luo & Tu, 2020). The max pooling layer of size 5 picks a single feature in each patch of a single encoded sequence. These extracted features then move to LSTM layer with 50 units, a recurrent drop out of 1% and an activation function of rectified linear unit (ReLU). On longer sequences, ReLU works better because of its linear partly linear function. The derived features are further passed to dense layers which return a predicted class label (0= inactive reference sequence/ no motif, 1 = active reference sequence/ presence of motif). Like the CNN, I use 5-fold cross validation to evaluate the model and will use the aggregated model on the test set. Finally, the models compile in the same way as in section 5.4.2.

The hybrid model performs with an accuracy of 75.25% with a loss of 0.17 on the training set (Figure 12. (B)). Although the hybrid model shows high loss for the accuracy, in comparison to the CNN however, the hybrid model might be marginally better. However, the high accuracy of the hybrid model could be due to the LSTM layer. Overall, the deep learning models perform well in classifying active and inactive sequences on the lymphoblastoid training set. However, it is yet to be seen if the models learning from lymphoblastoid is sufficient to identify real time active sequences from GTEx.

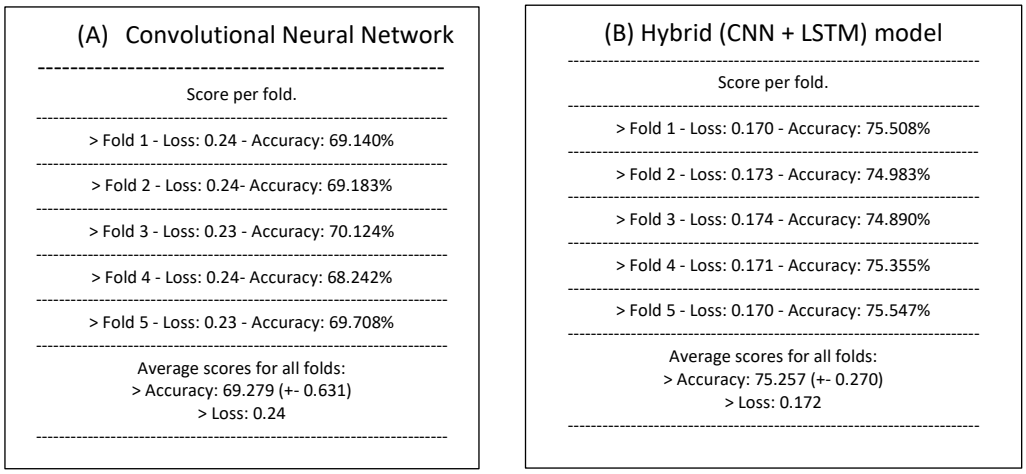


Figure 12. Performance of the CNN and Hybrid model on training data: The box (A) represents the performance of the CNN model. The CNN attains an average accuracy of 69.27%. The box (B) represents the average performance of the hybrid model. The hybrid model attains an average accuracy of 75.25% on the lymphoblastoid test data.

6.2.4 Model performance

In this section, we will first talk about the performance of the models on some ML metrics after which we will discuss about their performance on the GTEx and MinE data.

To determine the performance of the models we use the following ML metrics: *(1) Error Rate* is the inaccuracy of predicted output values on a categorical data. It can be understood as the proportion of cases where the prediction is wrong. *(2) Accuracy* is the number of correctly predicted data points among all the data points. *(3) Sensitivity* is a measure of the proportion of

actual positive cases that got predicted as positive (or true positive). Sensitivity is also called as *recall*. Higher the sensitivity, better the performance. *(4) Specificity* is defined as the proportion of actual negatives that got predicted as negative (or true negative). Higher the specificity, better the performance. *(5) Precision* is the fraction of relevant instances that were retrieved. Higher scores correspond to better performance.

$$Error\ Rate = \frac{\text{False Positive } + \text{ False Negative}}{\text{True } + \text{ False}}$$

$$Accuracy = \frac{\text{True Posiitive } + \text{True Negative}}{\text{True } + \text{ False}}$$

$$Sensitivity\ (Recall) = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Negative}}$$

$$Specificity\ = \frac{\text{True Negative}}{\text{True Negative } + \text{ False Positive}}$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Positive}}$$

$$False\ Positive\ Rate\ = \frac{\text{False Positive}}{\text{True Negative } + \text{ False Positive}}$$

$$F1\ Score = \frac{2\ (\text{Precision})\ (\text{Recall})}{\text{Precision } + \text{ Recall}}$$

$$Mathew\ Correlation\ Coefficient = \frac{(\text{TP}.\text{TN}) \ - \ (\text{FP}.\text{FN})}{\sqrt{(\text{TP} + \text{FP})\ (\text{TP } + \text{ FN})\ (\text{TN} + \text{FN})(\text{TN } + \text{ FN})}}$$

(6) F1 score and *Matthew correlation coefficient (MCC)* are like accuracy but are supposed to be more reliable. F1 score is the mean of the precision and recall whereas MCC is typically used in machine learning for bioinformatics (Chicco & Jurman, 2020). F1 score reaches its best value at 1 and worst value at 0. MCC is defined as the measure of the quality of binary classification. An MCC score of 1 indicates a perfect result. A score of 0 is expected for a prediction no better than random guess and a score of -1 indicates a total disagreement between prediction and observation. *(7) Area under curve (AUC)* is a measure of the ability of the classifier to distinguish between classes. Higher the AUC, the better the performance of the model at distinguishing two classes. *(8) Receiver operating characteristic curve (ROC)* is a graph that illustrates the diagnostic capability of a binary classifier at different discrimination thresholds. The diagonal in the ROC plot depicts random guess of the model (50%). The accuracy of a model is determined looking at the curve running over the random guess. The model is said to perform the opposite of what it should if the curve goes under the random guess line.

Performance of deep learning models

We determine the efficiency of the model by its accuracy in identifying mutations on true positive GTEx labels. The CNN model correctly predicts a total of 1.82% of the mutations on test set (size = 34, 667; (Figure 13. (A)). On the other hand, the hybrid model correctly predicts

only 1.39% of the mutations on the test set (Figure 13. (B)). Overall, we can see that although the models perform well on the training set in classifying active and inactive reference seqeunces, it is unable to identify active sequences form the GTEx and MinE. This may be because the subtle effect that the GTEx and MinE does not push them to either active or inactive. Therefore, the models are predicting the active sequences as inactive.

(A) Confusion Matrix of the CNN trained on Lymphoblastoid.

		ACTUAL	
		GTEx	MinE
PREDICTED	GTEx	632	603
	MinE	17247	16185

(B) Confusion Matrix of the hybrid trained on Lymphoblastoid

		ACTUAL	
		GTEx	**MinE**
PREDICTED	**GTEx**	485	386
	MinE	17394	16402

Figure 13. Confusion matrix of the CNN and Hybrid models trained on Lymphoblastoid and tested on GTEx and MinE: Boxes (A) and (B) show the confusion matrix. The CNN model was only able to identify 632 out of 17,879 and 603 out of 16,788 true labels from MinE. The hybrid model seems to perform worse when compared to the CNN, with 484 out of 17,879 and 384 out of 16,788 true labels from GTEx.

The CNN trained on active (motif) reference sequences and inactive reference sequences was able to predict 3.53% of GTEx mutations correctly among a total of 17,879 total GTEx motif. This means that the model correctly predicts 3.53% of the active motifs known to disrupt transcription in its TFBS thereby effecting gene expression at the mRNA level. The hybrid model trained on active and inactive reference sequences was able to predict 2.71% of GTEx mutations correctly.

The CNN model performs with an accuracy of 48.5% and an error rate of 0.515 on the test set (Figure 14. (B)). The F1 score of the model is 0.065 and the MCC of the model is 0.001. The sensitivity of the model is 0.035 and the specificity is 0.96 with a precision of 0.51. The AUC of the CNN is 0.4255 (Figure 14. (D)). The hybrid model performs with an accuracy of 48.7% and a high error rate of 0.513 (Figure 14. (A)). The F1 score of the model is 0.052 and the MCC of the model is 0.0132. The MCC of the model suggests that the hybrid model infers a bad performance. These metrics of the CNN and hybrid model suggest that the model performed was poor than a random guess. The sensitivity of the model is 0.027 and the specificity is 0.97 with a precision of 0.55. The AUC of the hybrid model is 0.4067 (Figure 14. (C)). Although, some of the metrics show slightly higher performance than the CNN, the overall performance of the hybrid model is bad as compared to the CNN. Therefore, the ML metrics indicate that both the hybrid model and CNN trained on lymphoblastoid have a very poor performance in classification.

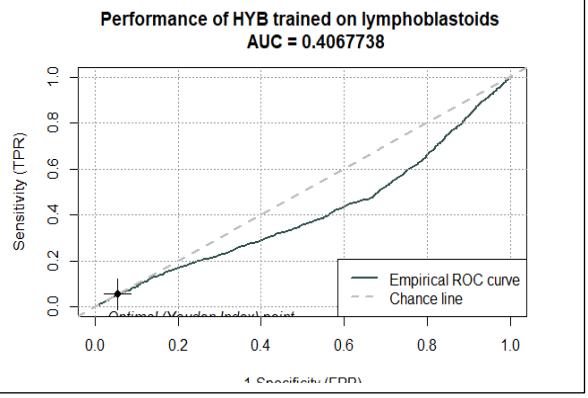
(A)Performance of hybrid trained on Lymphoblastoid.

Error Rate	*0.513*
Accuracy	*0.487*
Sensitivity	*0.0271*
Specificity	*0.977*
Precision	*0.556*
False Positive Rate	*0.023*
F1 Score	*0.05162*
Matthew Correlation Coefficient	*0.0132*

(B)Performance of CNN trained on Lymphoblastoid.

Error Rate	*0.515*
Accuracy	*0.485*
Sensitivity	*0.035*
Specificity	*0.964*
Precision	*0.512*
False Positive Rate	*0.036*
F1 Score	*0.0655*
Matthew Correlation Coefficient	*0.0015*

(C)



(D)

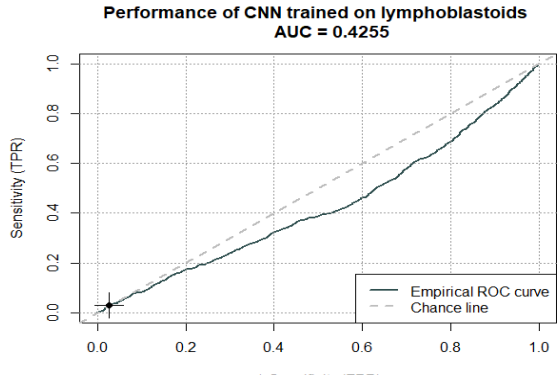


Figure 14. Performance metric of the CNN and hybrid model trained on Lymphoblastoid and tested on GTEx and MinE: Boxes (B) and (D) represent the various performance metrics used along with the confusion matrix to evaluate the performance of the CNN model trained on lymphoblastoid. The boxes (A) and (C) represent the performance of the hybrid model. The models perform worse that a random guess. In other words, the model quite does the opposite of what it should.

Models on Genotype Tissue Expression (GTEx)

The following Figure 15 shows the size of mutation as predicted by the CNN plotted against the effect size of the known GTEx mutations. We look if the size of the mutation is directly proportional with the effect size. The effect size is derived using the standard eQTL framework. In the following plot I look for changes that are greater than -1, 1 and that have a difference in change greater than 0.50. The negative sign indicates directionality which we ignore in this project.

The CNN model predicts that there are no mutations with the size of 0.75 and greater than -1 and 1 (Figure 15. (A)). However, there are few mutations which are greater than -1, 1 and mutation size of 0.50. The CNN model infers that the most significant mutations have a mutation size less than 0.25 on GTEx test data. Importantly, the ones having the strongest effect seem to have size of less than 0.10. Therefore, the CNN on GTEx infers that various types of mutations can occur with differing size. But the ones with the strongest effect have a size less than 0.50. On the contrary, the hybrid model seems to suggest that all mutations have a small size (less than 0.50; Figure 15. (B)). However, like the CNN, all the mutations suggested to have a very strong effect have the smallest size of less than 0.10. The hybrid model trained on lymphoblastoid infers that all mutations have a small mutation size but the ones with the strongest effect have an even smaller mutation size.

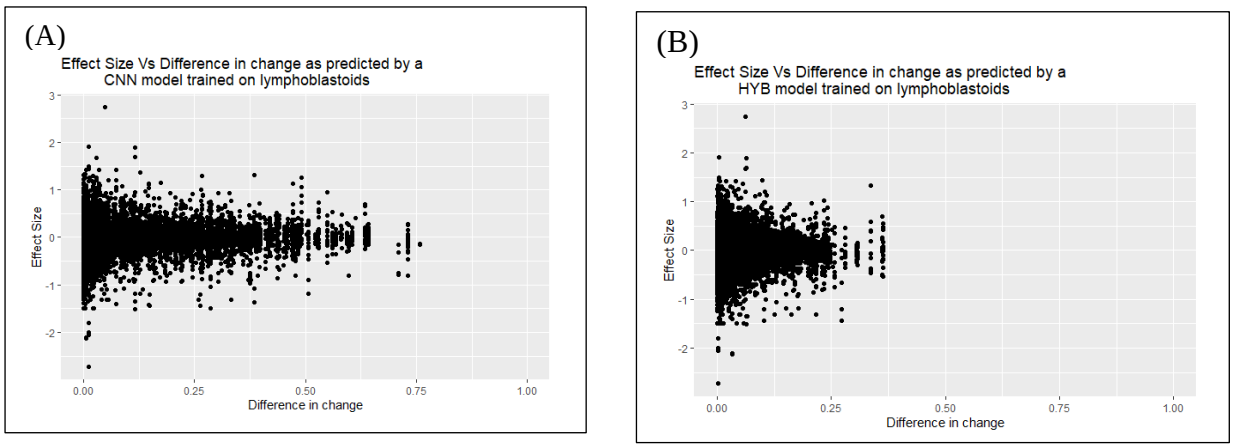


Figure 15. Comapring the effect size of the as per GTEx to the size of mutation predicted my the CNN and hybrid model: (A) The CNN model suggests that although there are some mutations with higher effect and bigger size. Most of the high effect mutations have a very small size. **(B)** The hybrid model suggest that all mutations occure within the size of 0.50. The rare mutations are lesser than 0.25

Models on Project MinE

The Figure 16 represents the prediction of the size of mutation from the CNN model plotted against phred scores from the CADD file. As discussed in the Section 6.2.2 the mutations with a score greater than 10 represents that such mutations are in the 10% of all mutations possible within the human reference genome. Scores greater than 20 belong to the top 1% of the mutations possible in the reference genome. There are only few mutations with greater size and that lie in the top 10% of the mutations possible in the human reference genome. The model infers that most of the mutations that are in the top 1% or even more rare have a size less than 0.50. However, the CNN predicts that most significant mutation within the range of phred score 10 to 40 are within the mutation size of less than 0.25.

Unlike the CNN on MinE, the hybrid model seems to suggest that all possible mutations in the top 10 % lie in a range of 0.25 – 0.50. The other mutations that lie in the top 1% of all possible mutations in the human reference genome have a size of less than 0.25. Therefore, the hybrid model trained on lymphoblastoid infers that, all the mutations possible within the human reference genome including the ones possibly disrupting the TFBS have a size less than 0.50. But the mutations with the strongest effect have a very low size less than 0.25. We see a similar phenomenon of hybrid model on MinE data as on GTEx. However, the hybrid model on project MinE data suggests some significant mutations in the top 10% and a couple in the top 1% having relatively a bigger size of mutations, but less than 0.50. However, the hybrid model seems to suggest like CNN that most mutations having the strongest effect have the size of less than 0.25 on the MinE test data.

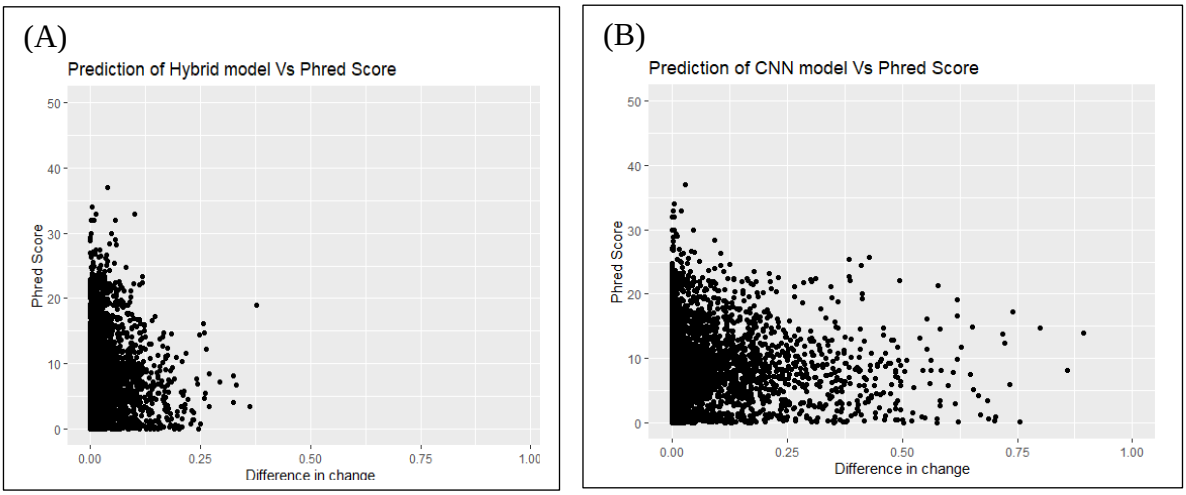


Figure 16. Comparing the CADD based phred scores to the size of the mutation as predicted by the CNN and Hybrid model. (A) Similar to GTEx the hybrid model suggests that all mutations occur with a size of less than 0.50 with the rare mutations less than 0.25. **(B)** Like on GTEx, the CNN suggests that although there might be some mutations with a bigger size and higher effect exist, most of the mutations are less than 0.25

6.2.5 Summary

In this section we train the models on the active motifs and inactive reference sequences from lymphoblastoid. we train and test two deep learning models CNN and hybrid (CNN + LSTM).

On GTEx, the deep learning models infer that not all mutations with big effect size have big size of mutation. Although there might be some of them, the CNN suggests that most of the mutations with higher effect have a very small size of mutation. The hybrid model on the other hand infers that all the mutations that happen have small size of mutation and the mutations with big effect size have a very small size of mutation. The general inference of both these models on GTEx data is that the effect size is inversely proportion to the size of mutation. Therefore, based on the performance metrics, we can infer that the deep learning models trained on lymphoblastoid does not efficiently identify mutations affecting gene expression due to the very low accuracy.

On project MinE, we also see a similar inference as on GTEx. The CNN model infers that mutations happen in all sizes, but the most significant ones (1% or rare) have the smallest size of less than 0.25. However, the hybrid model seems to suggest that all the likely ALS mutations have a small less of less than 0.50. The ones having the highest effect size (or the mutations known to lead to an aberrant regulation of protein) have the lowest size of mutations less than 0.10. Therefore, the CNN and hybrid model having trained 140,865 and tested on 34,667 active and inactive reference sequences in the lymphoblastoid infer than bigger effect of the mutation is inversely proportional to the size of mutation. In view with the performance metrics both the model architectures trained on lymphoblastoid do not do a good job in identifying mutations in GTEx and MinE.

6.3 Regions around the transcription start site in lymphoblastoid.

In this section, we zoom in and check if the performance of the models changes if we train the models on the regions around the transcription start site within the lymphoblastoid. My idea in this section is to train the models on regions where it is known that mutations have the strongest effect. These regions are the transcription start site (TSS). Transcription start site regions are where the process of transcription is initiated when a certain TF binds with the DNA. Therefore, mutations in this region have the strongest effect. Training the deep learning models on the DNA sequences from TSS should improve the ability of the model to clearly distinguish the GTEx mutations from MinE since the training set comprises of motifs that can effect transcription. The test used in the section is the same as in Section (6.2).

6.3.1 Training Set

The raw file *TSS_GRCh37.75.10kb.bed.gz* comprises of all possible mutations in transcription start site throughout the human reference genome. We need to train the model on TSS regions in the lymphoblastoid. To do so, we first intersect TSS raw file with the lymphoblastoid raw file from the above section using bedtools intersect. Similar as in the above section 6.2.1, we use the same process to generate active and inactive reference sequences length 150. RMSE between a pair of active and inactive reference sequences is less than 2%. Therefore, to train the models on regions around TSS, we have a balanced data set of size of 189,791 active and inactive reference sequences extracted from the regions around TSS within the lymphoblastoid. These sequences are further one-hot-encoded as in Section 5.4.1 to train the model.

6.3.2 Test Set

The test set is same as the test set for the lymphoblastoid models (Section 6.2.2). The complete test set involves 34,667 samples including 17,879 GTEx and 16,788 MinE sequences along with their corresponding effect size (for GTEx) and phred score (for MinE).

6.3.3 Model architecture

The model architecture is the same as above (Section 6.2.3; Figure 7). I train both my model architectures on 1,89,791 active and inactive reference sequences. The average accuracy across the 5 folds for the CNN model is 69.57% with a validation loss of 0.22 on the training set. The CNN model performs consistently in every fold (Figure 17. (A)). Similarly, the hybrid model also performs with an average accuracy of 75.02% with a validation loss of 0.17 on the training set (Figure 17. (B)). Like in lymphoblastoid model, the hybrid model (Figure 8) trained on TSS has higher accuracy in the classification than the CNN. However, having trained on TSS both the models exhibit high loss in comparison to the model accuracy.

6.3.4 Model performance

The CNN model predicts 920 GTEx true positives on a test set of size 34,667. This means that the CNN model correctly identifies 2.65% of the mutations effecting gene expression on the complete test set (Figure 18. (B)). The CNN trained on active motif reference sequences and inactive reference sequences was able to predict 5.14% of GTEx mutations correctly (Figure 22. (A)). On the other hand, the hybrid model correctly identifies 1.35% of the mutations effecting gene expression (Figure 18. (A)) on the test set. The hybrid model trained on active

motif reference sequences and inactive reference sequences was able to predict 2.63% of GTEx mutations correctly within known GTEx motifs in the GTEx file.

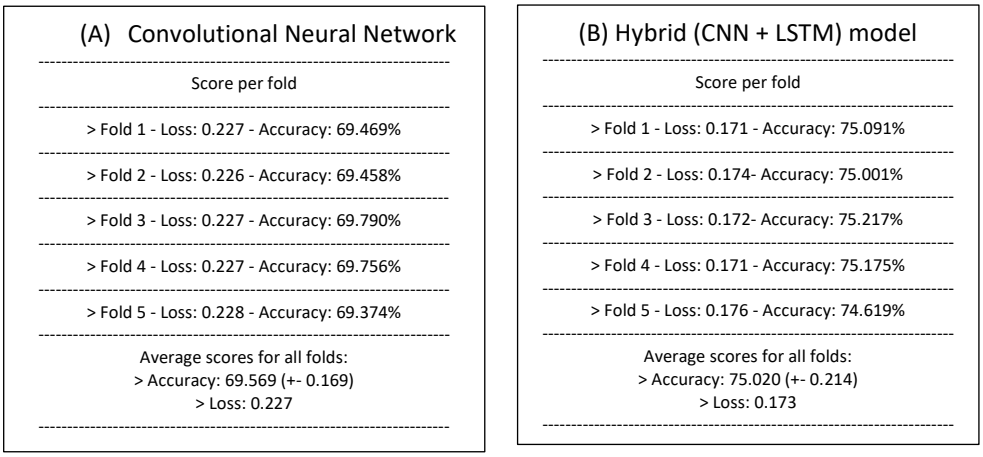


Figure 17. Performance of the CNN and Hybrid model on trained on TSS: The box (A) represents the performance of the CNN model. The CNN attains an average accuracy of 69.56%. The box (B) represents the average performance of the hybrid model. The hybrid model attains an average accuracy of 75.02% on the lymphoblastoid test data.

The CNN model performs with an accuracy of 48.8% and an error rate of 0.511 (Figure 19. (B)). The F1 score of the model is 0.093 and the MCC of the model is 0.001. These metrics indicate that the CNN model trained on TSS performs very poorly. The sensitivity of the model is 0.051 and the specificity is 0.95 with a precision of 0.54. The AUC of the CNN is 0.4226 (Figure 19. (D)), which is the same as the CNN trained on lymphoblastoid (Figure 14. (D)). The hybrid model on the other hand performs with an accuracy of 48.7% and an error rate of 0.512 (Figure 19. (A)) on the test set. The F1 score of the hybrid model is 0.093 and the MCC of the model is 0.019. The sensitivity of the model is 0.026 and the specificity is 0.979 with a precision of 0.57. The AUC of the CNN is 0.423 (Figure 19. (C)). These metrics of the CNN and hybrid models suggest that my models underperform in predicting mutations that effect gene expression.

(B) Confusion Matrix of the hybrid model trained on TSS within Lymphoblastoid

		ACTUAL	
		GTEx	MinE
PREDICTED	GTEx	471	344
	MinE	17406	16444

(B) Confusion Matrix of the CNN trained on TSS within Lymphoblastoid

		ACTUAL	
		GTEx	**MinE**
PREDICTED	**GTEx**	920	784
	MinE	16959	16004

Figure 18. Performance of the CNN model trained on TSS within Lymphoblastoid and tested on GTEx and MinE: Boxes (A) and (B) represent the various performance metrics used along with the confusion matrix to evaluate the performance of the CNN model trained on TSS within lymphoblastoid.

Models on Genotype Tissue Expression (GTEx)

These results are like the CNN trained on lymphoblastoid (Part A). The CNN model predicts that there are no mutations with the size of 0.75 and greater than -1, 1. Unlike the CNN trained on lymphoblastoid, there are few mutations which are greater than -1, 1 between the mutation size of 0.50 – 0.75 (Figure 20. (B)). This could mean that the newer TSS data provided in some ways is improving the model performance. However, the ML metrics say otherwise. The CNN model trained on TSS infers that the most significant mutations (greater than -1 and 1) have a mutation size less than 0.50 and some mutations with a mutation size less than 0.75 on GTEx test data. The ones having strong effect vary in different sizes. But the most significant ones are still less than 0.25.

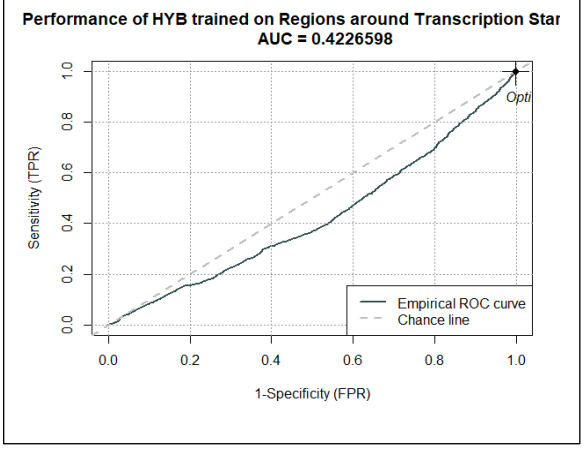
(A)Performance of hybrid trained on TSS within Lymphoblastoid.

Error Rate	*0.512*
Accuracy	*0.487*
Sensitivity	*0.026*
Specificity	*0.979*
Precision	*0.578*
False Positive Rate	*0.047*
F1 Score	*0.0937*
Matthew Correlation Coefficient	*0.0197*

(B)Performance of CNN trained on TSS within Lymphoblastoid.

Error Rate	*0.511*
Accuracy	*0.488*
Sensitivity	*0.051*
Specificity	*0.953*
Precision	*0.54*
False Positive Rate	*0.047*
F1 Score	*0.0931*
Matthew Correlation Coefficient	*0.0110*

(C)



(D)

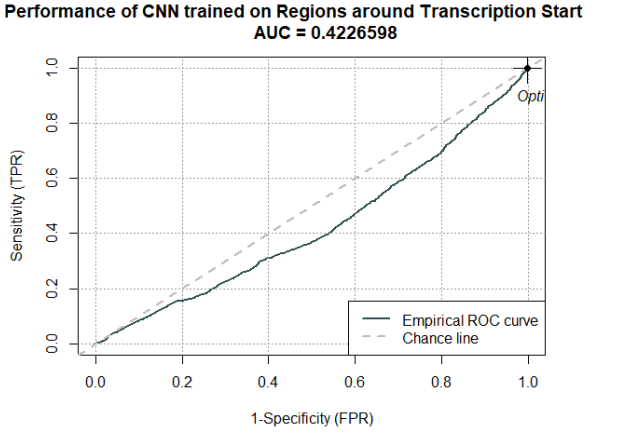


Figure 19. Performance metric of the CNN and hybrid model trained on Lymphoblastoid and tested on GTEx and MinE: Boxes (B) and (D) represent the various performance metrics used along with the confusion matrix to evaluate the performance of the CNN model trained on lymphoblastoid. The boxes (A) and (C) represent the performance of the hybrid model. The models perform worse that a random guess. In other words, the model does the opposite of what it should.

The (Figure 20. (A)) shows the size of mutation as predicted by the hybrid model plotted against the effect size of the known GTEx mutations in the training set. These results are like the hybrid trained on lymphoblastoid (Part 1). The hybrid model predicts that all mutations have a small size of mutation. But the ones with strongest effect are less than or equal to 0.25. There are no mutations with the size of 0.75 and greater than -1 and 1. The hybrid model trained on TSS infers that the most significant mutations (greater than -1 and 1) have a mutation size less than 0.25 and mostly less than 0.10 on GTEx test data. However, based on the performance

metrics I can conclude that the model does not perform efficiently in identifying mutations by disrupting TF's and affecting gene expression.

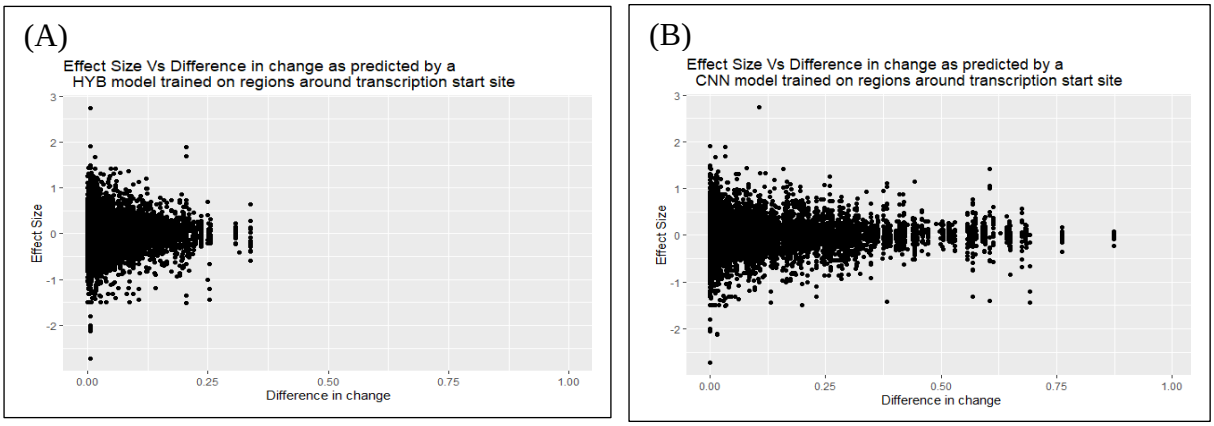


Figure 20. Figure 16. Comparing the GTEx effect size with the size of the mutation as predicted by the CNN and Hybrid model. (A) Similar to section 6.2 the hybrid model suggests that all mutations occur with a size of less than 0.50 with the rare mutations less than 0.25. **(B)** The CNN suggests that although there might be some mutations with a bigger size and higher effect exist, most of the mutations are less than 0.25

Models on Project MinE

The Figure 21 represents the prediction of the size of mutation predicted by deep learning models plotted against phred scores from the CADD file. CADD consist of scores that are assigned to the mutations observed in ALS patients. Mutations with a score greater than 10 represents that they are in the 10% of all mutations possible in the human reference genome. Scores greater than 20 belong to the top 1% of the mutations possible in the reference genome.

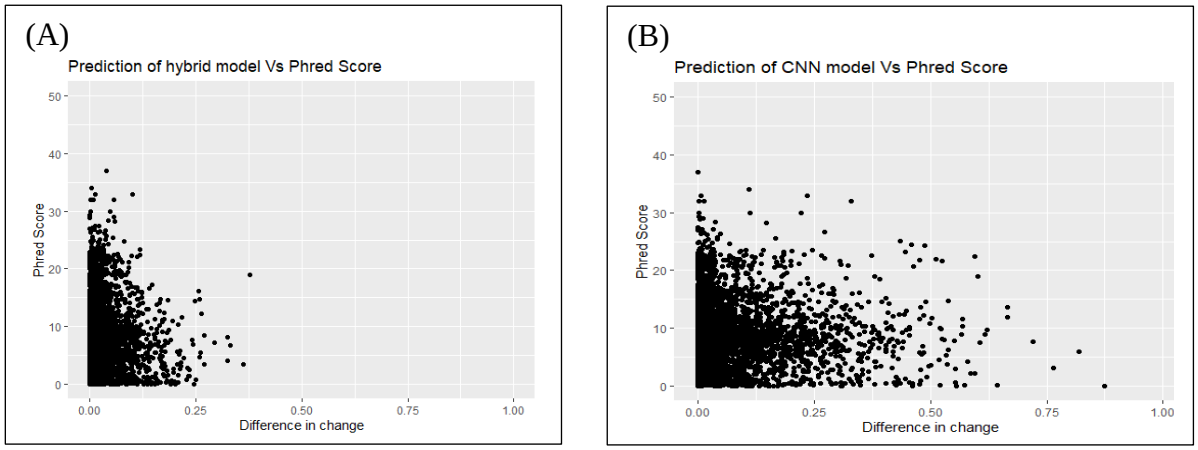


Figure 21. Comparing the CADD based phred scores to the size of the mutation as predicted by the CNN and Hybrid model. (A) Similar to GTEx the hybrid model suggests that all mutations occur with a size of less than 0.50 with the rare mutations less than 0.25. **(B)** Like on GTEx, the CNN suggests that although there might be some mutations with a bigger size and higher effect exist, most of the mutations are less than 0.25

There are only few mutations as predicted by CNN with greater size and that lie in the top 10% of the mutations possible in the human reference genome (Figure 21. (B)). Unlike the CNN trained on lymphoblastoid, we see quite some mutations with a size of less than 0.75. As the size of the mutation decreases, the effect of it seems to increase. However, like the CNN trained on lymphoblastoid, most significant mutations causing the highest effect are still less than 0.25.

We see no difference in performance of the hybrid model trained on TSS as compared to the hybrid model trained on lymphoblastoid (Figure 21. (A)). Like the hybrid model trained on lymphoblastoid, the hybrid model trained on TSS predicts that all possible ALS mutations have a small size and the ones with higher effect have an even smaller mutation size. Therefore, trained on the non-coding DNA sequence comprising of motifs from TSS in lymphoblastoid, the hybrid model infers that the most significant and the most insignificant mutations all have a size of less than 0.25. Therefore, both models infers that most of the mutations that are rare and have highest effect in the human reference genome have a mutation size less than 0.25. Overall, like the models on GTEx the performance metrics suggests that my deep learning models do not do an efficient job in identifying mutations that destroy TFBS thereby leading to ALS.

6.3.5 Summary

In this section we train my model on the active motifs and inactive reference sequences from the regions around TSS within the lymphoblastoid. We do this to check if the model improves its efficiency in identifying active sequences from GTEx if we provide the DNA sequences which comprise of regions where the transcription is initiated. We see a change in performance among the models trained on lymphoblastoid and TSS. We see that, on the test set the model is now better to classify active GTEx mutations from the MinE mutations. However, this is no significant change.

On GTEx, the models suggest that there are no significant mutations and although there might be some of them, the CNN suggests that most of the mutations with higher effect have a very small size of mutation. Just like in Part (A), The hybrid model infers that all the mutations that occur have small size of mutation and the mutations with big effect size have a very small size of mutation. The general inference of both these models on GTEx data is that the effect size is inversely proportion to the size of mutation. However, the performance metrics of the models show that both the hybrid and CNN model underperform in identifying significant mutations that destroy gene expression.

On the project MinE data however, CNN suggests that some significant mutations having higher effect have a big size of mutation. On the contrary, the hybrid model infers that all the ALS motifs mutate with a small size. However, both models are similar in a way that they suggest that mutations causing the most significant effect have a size less than or equal to 0.25. Like on GTEx, the deep learning models underperform in identifying mutations disruption the TFBS.

Having observed the performance of both the models on blood and their performance of GTEx and MinE test data, we can determine up to a large extent that the deep learning models would also underperform in identifying MinE mutations when trained on the lower motor neuron data. This is because of the underlying biological complexity associated with the organ.

We can say so because throughout this section, we train our model architectures on specific cell data in blood (lymphoblastoid and TSS). We see that my model architectures do not perform well in identifying mutations when trained on such data. The lower motor neuron is a combination of motor neurons made up of millions of cells and therefore leads to more complexity associated with it. So, we can expect the model to perform with a very low accuracy on the training set and the test dataset comprising of ALS mutations from Project MinE.

Therefore, in the next section Part 2, I train my model architectures on the DNA sequences from the lower motor neuron. I test up to what extent can both the architectures having trained on lower motor neuron can predict active motifs in the *promotor* region of ALS patients. Promotor regions comprise of short DNA sequences to which proteins bind and initiate the transcription. This will let us know about the accuracy with which our models perform to determine motifs that might lead to the aberrant protein regulation subsequently leading to ALS. I test if we can determine the genes involved within which these mutations lead to drastic effect.

6.4 Part 2: Models on lower motor neuron

6.4.1 Training set

The raw file *Song_Regnetwork_astro.bed* (Song et al, 2019; Figure (22)) consists of active regions in the lower motor neuron. The author of the paper annotates relationship between and regulatory elements in cell types that are relavant to complex neuropsychiatric disorders.

The raw file comprises of the chromosome number, its start and end position. It also consists of the gene corresponding to start and end positions and its type if it is a promotor or an enhancer. A promotor is a DNA sequence which turns a gene on or off. The process of transcription is initiated at the promoter. Typically found at the beginning of the gene, the promoter has a binding site for the enzyme used to make a messenger RNA (mRNA) molecule. An enhancer on the other hand is a short (50 – 1500 base pair) region of DNA that can be bound by proteins to increase the likelihood of transcription in the gene. These are also sometimes referred to as transcription factors (TF).

We generate a normalised training set as in section 6.2.1 of size 240,000, half of which comprise of motifs in non-coded DNA sequences and the RMSE between a pair of active and inactive sequence is less than 2%.

chrom	chromStart	chromEnd	gene	porE
1	884689	904689	NOC2L	promotor
1	903641	927394	RP11-5407.17	enhancer
1	907497	927497	C1orf170	promotor
1	923431	943431	RP11-5407.17	promotor
1	927395	936954	MIR200B	enhancer
1	943677	957199	TNFRSF18	enhancer
1	1041741	1061741	C1orf159	promotor
1	1092484	1112484	MIR200B	promotor
1	1132071	1152071	TNFRSF18	promotor
1	1157411	1177411	SDF4	promotor
1	1199265	1219265	UBE2J2	promotor
1	1206874	1212438	DVL1	enhancer
1	1233947	1253947	PUSL1	promotor
1	1274730	1294730	DVL1	promotor
1	1287157	1307157	MXRA8	promotor
1	1300875	1320875	AURKAIP1	promotor

Figure 22. Raw file Song_Regnetwork_astro.bed: These comprise of chromosome start and end position, the corresponding gene and its information if it is a promoter or an enhancer. These regions comprise of motifs among the sequences associated with complex neuropsychiatric diseases. I use this file to extract the training set for the model.

6.4.2 Test Set

To compare the performance of the model, we use CADD scores. We use the same procedure as in 6.2.2 to generate a test set comprising of 50,000 mutations in the promoter regions of ALS

in project MinE data (Project MinE ALS Sequencing consortium, 2018) and their corresponding phred scores (Figure 23). Because we train the model on non-coded DNA sequences from the lower motor neuron intrinsic to complex metrocentric diseases, we only attempt to identify up to what extent is training the deep learning model on this kind of data set able to identify mutations known to cause ALS. We do not have other true positive label such as GTEx to evaluate the reliability of the model. Thereby this section will not comprise of other ML metrics.

884730	884731	G	A	-0.129573	1.148
884755	884756	G	A	-0.624553	0.035
884760	884761	C	T	-0.170979	0.868
884766	884767	G	A	-0.329071	0.298
884810	884811	G	T	-0.138344	1.082
884814	884815	A	G	-0.105238	1.353
884823	884824	G	A	-0.106461	1.342
884829	884830	C	T	-0.207947	0.678

Figure 23. Combined Annotation -Dependent Depletion (CADD; Rentzsch et al, 2019): The CADD file comprises of the reference and altered nucleotide along with their corresponding chromosome start and end position. The file comprises of two rows phred and raw scores. The last column corresponds to the phred scores. The phred scores are log transformed raw scores.

6.4.3 Model architecture

The model architecture is the same as above (Section 6.2.3). I train both my model architectures on 240,000 active and inactive reference sequences. The RMSE between the active and inactive reference sequences is less than 2%. The average accuracy across the 5 folds for the CNN model is 56.67% with a validation loss of 0.34 on the training set (Figure 24. (A)). Similarly, the hybrid model performs with an average accuracy of 60.87% with a validation loss of 0.23 on the training set (Figure 24. (B)). There is not much significant difference between the CNN and the hybrid modem based on the performance on the training set and both the models have a high loss compared to the accuracy of the model.

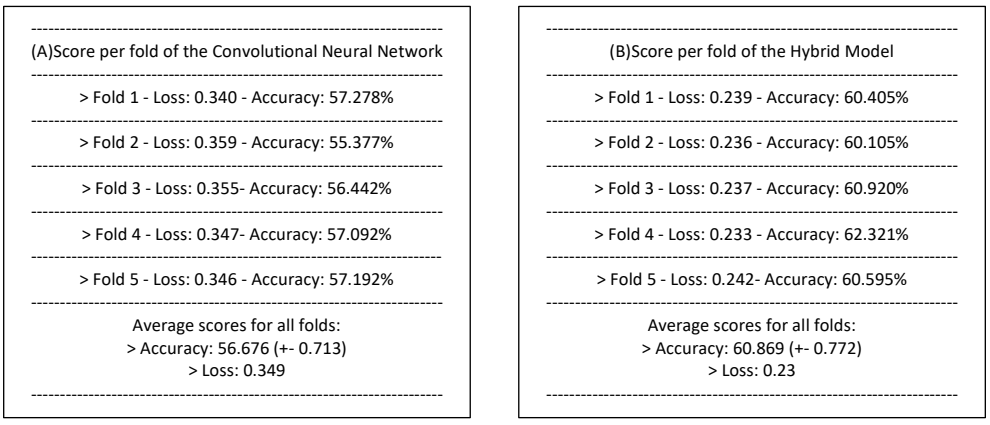


Figure 24. Performance of the CNN and Hybrid model on trained on promoters and enhancers in lower motor neuron: The box (A) represents the performance of the CNN model. The CNN attains an average accuracy of 56.67% and a high loss of 0.34. The box (B) represents the average performance of the hybrid model. The hybrid model attains an average accuracy of 60.86% and high loss of 0.24 on the lymphoblastoid test data.

6.4.4 Model performance

Models on Project MinE

Unlike the CNN models trained on blood we see a different result. In the previous sections, the CNN models trained on blood seem to suggest that only few mutations have a mutations size greater than 0.50 and as the size of the mutation decreases its effect size increases both in GTEx and MinE data. In both the plots (Figure 25) we see that no two mutations within a single gene have the same size of mutation or even the same effect of mutation.

However, the CNN model trained on promoters and enhancers in the lower motor neuron seems to suggest that mutations come in all sizes and effects. In the Figure 25. (A), we see diverse size of mutations compared against the significance of the mutation. This phenomenon could however be explained. This performance of the CNN could be attributed to the training set. We train the model on active and inactive reference sequences that are related to complex neuropsychiatric disorders. Therefore, the comorbidities among these neuropsychiatric diseases could be a reason for the CNNs prediction for the size of ALS mutations in the promoter regions.

The hybrid model seems to be consistent in its performance on predicting the size the mutations throughout all datasets (Figure 25. (B)). Having trained on complex data related to neuropsychiatric disorders the model seems to suggest that all ALS mutations across species have a very small size of mutation and the rare mutations have an even smaller size of less than 0.25. Although we see contrasting results by comparing both the models, the general inference to a large extent is that most mutations likely to cause ALS are of size less than 0.25 irrespective of familial or sporadic ALS.

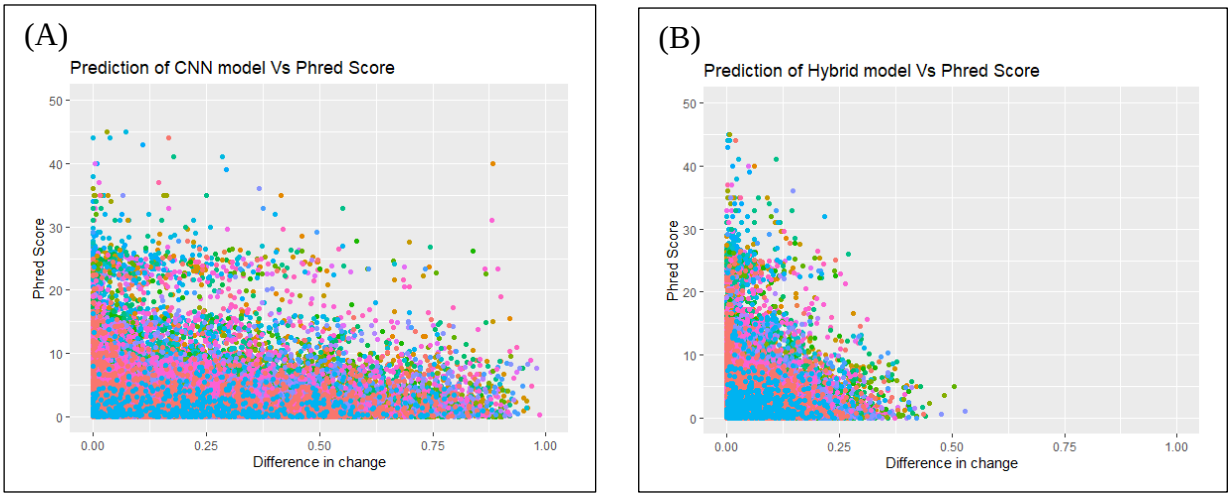


Figure 25. (A) and (B) show the CADD based phred scores compared to the size of the mutation as predicted by the CNN trained on lower motor neuron: (A) The CNN model trained on non-coding sequences in LMN suggests that ALS mutations may occurs in all the sizes. (B) The hybrid model suggests that all mutations are less than 0.25. However, the most important mutations are less than 0.25.

6.4.5 Summary

Having trained both the deep learning architectures on the active promoter and enhancer sequences within the lower motor neuron, we see similar results as in the above sections for

the hybrid and a different phenomenon compared to CNN. However, both the CNN and hybrid underperform.

On Project MinE data, both the models perform differently. The CNN model infers that all the mutations in active motifs associated with ALS come in different sizes and effects. We see high effect and rarest mutations ranging between 0 to 1. However, most mutations that have the highest effect still lie in the range of 0 to 0.50. The CNN model was able to predict 12.39% of the MinE mutations correctly that that disrupt the TFBS. In contrast, the hybrid model shows us a similar inference like seen on blood cells. The hybrid model infers that all the mutations that occur in the body have a small size of mutation. However, the mutations with the highest effect are less than 0.25. The hybrid model correctly classifies only 6.40% of project MinE motifs known to disrupt the TFBS that lead to aberrant protein regulation.

However, both the models CNN and hybrid do not perform well provided the test set size of 50,000. Therefore, models trained on lower motor neuron cannot efficiently identify ALS mutations due to the biological complexity involved, inefficiency of my deep learning models to identify mutations disrupting TFBS and most probably in the training procedure.

7. Result

In this project we attempt to identify mutations disrupting TFBS thereby leading to ALS using deep learning. To do so, I first establish the efficiency of the deep learning models in identifying mutations that disrupt the transcription process. We use GTEx as the true label to test my models to identify mutations that disrupt transcription which effects gene expression. We also use the Project MinE (Project MinE ALS Sequencing consortium, 2018) data as false labels to identify mutations that disrupt the transcription that leads to disruption of TFBS ultimately leading to aberrant protein synthesis. We see that both the model architectures underperform in identifying mutations and show different kinds of predictions on the GTEx and MinE data.

The CNN trained on lymphoblastoid identifies 3.59% of the MinE mutations and 3.53% of the mutations effecting gene expression (GTEx). Similarly, the CNN trained on TSS identifies 4.67% of the MinE mutations (Project MinE ALS Sequencing consortium, 2018) and 2.63% of the GTEx mutations. Finally, the CNN trained on lower motor neuron predicts only 4.18% percent of the MinE mutations. These scores and ML metrics taken together suggest that the CNN model underperforms in the classification task. Overall, all the CNN models on MinE data infer that in ALS, mutations have different sizes with varying effects. But the mutations having the highest effect have the lowest size of mutation. However, this cannot be concluded due to the underperformance as shown by the ML metrics.

Similarly, the hybrid model trained on lymphoblastoid identifies only 2.29% of the MinE mutations and 2.71% of the mutations affecting gene expression. Similarly, the hybrid model trained on TSS identifies 2.04% of the MinE mutations and 4.38% of the GTEx mutations. Finally, the hybrid model trained on lower motor neuron predicts only 4.18% percent of the MinE mutations. These scores and ML metrics taken together suggest that the hybrid model underperforms in the classification task. The hybrid models consistently infers that any mutation GTEx or MinE have a low size of mutation and the ones with lower than a size of 0.25 have a very high effect size.

Both the CNN and Hybrid models suggest that although there might be very few of such mutations with big size of mutation proportional to the effect of mutation, most of the mutations

with a high effect have a very low size of mutations. However, this cannot be concluded because as per the performance metrics of deep learning the models is poor. In this project, the proposed deep learning architectures fail to identify mutations. This could mean: (1) the model must have missed learning something crucial in the training dataset. (2) The model architectures itself might not be deep enough to learn the complexity associated with the training set.

8. Discussion

There are two main reasons for the underperformance of my models. (1) Most successful deep learning models adopt two different approaches in the training procedure. (1.1) Using variable length sequences increases the reliability of the deep learning models instead of fixed length sequences. Using variable length sequences replicates the structure of DNA sequences in our body as DNA sequences in different chromosomes have different length (Zhou & Troyanskaya, 2015). (1.2) k-mer sequences are substring of a DNA sequence of length k. For example, a sequence 'ATGG' comprises of two k-mers 'ATG' and 'TGG'. However, all the models that perform well use one-hot encoding to encode the sequences. (1.3) In addition, it is also important to acknowledge the complexity associated with fixed length training sequences. The training sequences we use are high throughput sequences that try combinations of TFs to bind with them. This lets a deep learning model understand the complexity involved provided there are sufficient layers for the model to help learn. However, other type of sequences which are generally used by simpler models are only capable of binding to a couple TFs among the large of available TFs.

(2) The simpler networks observe high performance accuracy when trained on sequences bound by a single TF to identify a single motif mutation. For example, the DeepBind (Alipanahi et al, 2015) architecture is very much like mine. It uses sequences of variable length which pass through a single layer of convolutional layer that acts like a motif detector, a max-pool computes the maximum and average of each motif detectors response and a dense layer. This gives DeepBind an AUC greater than 70. On the other hand, the more complex models such as DeepSEA (Zhou & Troyanskaya, 2015) use 8 convolutional layers and it trains on 17% of high throughput active sequences from the human reference genome. The test comprises of large number of DNA sequences that can bind with any or all the approximately 1500 TFs in the human reference genome. This along with deeper layers lets DeepSEA identify large number of motif mutations within the human reference genome.

My models have relatively a simple network like DeepBind and it might do the job if we train the models on sequences that can be bound by a single TF resulting in very few motif mutations. However, the sequences used to train my model can bind by any or all the TFs available in the human reference genome. Thus, my simple networks find it difficult to capture the complexity associated with the training set due the shallow structure. Therefore, my models do not perform well in identifying motif mutations effecting gene expression or evening mutations disruption the TFBS. However, increasing the number of layers for the CNN along with novel expectation pooling (Luo et al, 2020) to pool the extracted features using expectation maximization. Maximization helps the models learn motifs on long sequences (greater than 100bp). Similarly, increasing the number of LSTM layers in proportion with the CNN with an increase in the recurrent dropout rate may lead to significant improvement in the model performance (Zhou & Troyanskaya, 2015).

9. Further work

On March 2020, Xio and Xinming introduced a novel pooling method named *'expectation pooling'* (Luo et al, 2020) within the deep learning framework rather than a maxpooling layer for predicting DNA-protein binding.Expectation pooling is divided into two sublayers, a local maxpooling layer and a desnse layer without additional hyperparameters. Their method improves the performance of the CNN when compared to using global maxpooling.

For the CNN, we could use six stacked convolutional layers with increased dialation and padding. To train the model on motifs in the non-coding sequences we will use k-mers along with including the expectation pooling instead of global max-pooling without. Using a large data set of k-mers and deep stacked networks with a dropout of 20% after every two layers should increase the performance of the models. Similarly, for the hybrid model we would use 4 stacked convolutional layers along with 3 LSTM layers. The combination of CNN+LSTM has been proven to work on long seqeunces instead of k-mers. Unlike my model, I would increase the recurrent dropout to upto 15% in every layer along with the number of units to upto 100. Making the mentioned upgrades should provide a deeper insight into the disease mechanism of ALS. With this architecture, the traing set we use which comprises of active DNA reference sqeunces that can bind with any or all the TFs is a good fit to train the improved model.

However, from an other perspective, the proposed simple model might work to identify small number of motifs if we train the models on specific sequences that can only bind with a single TF. This will enable us to identify context specific motif in a certain region. However, it is more significant to identify more number of motifs that can disrupt the TFBS. Therefore, probably an improved model is better to train on large number of squences.

10. Conclusion

In this project, we attempt to indentify such motif mutations which dirsrupt the TFBS leading protein aggregation in the surrounding regions and ultimately ALS. We build two deep learning models that can classify active and inactive reference DNA seqeunces. I then train the models on lymphoblastoid. Mutations in lymphoblastoid are known to effect transcription. We subseqeuntly train my model on such regions (TSS) in lymphoblastoid in which mutations have strongest effect because that is where the transcription process is initiated. This is to give the models a robust data to train on. We test my models on (1) GTEx which comprises of some motif mutations which may effect transcription. (2) Project MinE (Project MinE ALS Sequencing consortium, 2018) comprises of mutations as observed in ALS patients. Evaluating the models on the test set shows that both the CNN and hybrid model underperform in predicting mutations in GTEx and MinE that disrupt transcription. To benchmark the models, we use scores that determine effect size of a mutation compare it with the size of mutation as predicted by the model. This shows that a large part of the damage causing mutations that desrupt transcription have a very small size of mutation. However, this cannot be concluded because of the low performace of my models.

Finally, we train the models on non-coding DNA sequenes comprising of motifs in lower motor neuron correspondig to complex neuropsychiatric diseases. We do so to test the efficiency of the model in recognizing mutations that desrupt TFBS thereby leading to ALS. If the models performed better than it did, we could have used the models to predic mutations that cause

alzheimers or parkinsons since their pathological hallmark is also protein aggregation. However, these models infer that although some ALS causing mutations vary in size, a large part of damage causing mutations have a very low size of mutation. However, this cannot be concluded because of the inefficient model architecture and the training procedure which resulted in low performance metrics of the models.

The low performance of my model is mainly because of the model architectures and the training methodoloy. Humans comprise of thousands of motifs. The more rescent succesful models adopt complex deep neural netoworks (Zhou & Transkei, 2015) and train on large number of active sequences that can bind with any or all of the thousands of TFs in the human genome to identify large number of motif mutations that disrupt TF's. On the contrary, the simpler models train on sequences which are bound by a single TF to identify a small number motif mutations that disrupt the TF (Alipanahi et al, 2015). However the draw back of my model is that the model architecturs and the training procedure. Although a simpler network such as mine might do fine in identifying a single motif, the sequences that I train on are too complex for my simple model to absorb therefor underperforming on the test set. Improving the model architecture as proposed in section 8 will enable the enhanced model provided we use the same training set.

References

1. Aaron R. Quinlan, Ira M. Hall, BEDTools: a flexible suite of utilities for comparing genomic features, *Bioinformatics*, Volume 26, Issue 6, 15 March 2010, Pages 841–842, https://doi.org/10.1093/bioinformatics/btq033.

2. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2014). DNA, chromosomes, and genomes. Molecular biology of the cell, 184-185.

3. Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, *33*(8), 831-838.

4. Beer, M. A., & Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell*, *117*(2), 185-198.

5. Brown RH, Al-Chalabi A. Amyotrophic lateral sclerosis. Prog Med Chem. 2017;58:63–117.

6. Buenrostro, J. D., Wu, B., Chang, H. Y., & Greenleaf, W. J. (2015). ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Current protocols in molecular biology*, *109*(1), 21-29.

7. Byrne, S., Walsh, C., Lynch, C., Bede, P., Elamin, M., Kenna, K., ... & Hardiman, O. (2011). Rate of familial amyotrophic lateral sclerosis: a systematic review and meta-analysis. Journal of Neurology, Neurosurgery & Psychiatry, 82(6), 623-627.

8. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1-13.

9. Choong, A. C. H., & Lee, N. K. (2017, November). Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method. In *2017 International Conference on Computer and Drone Applications (IConDA)* (pp. 60-65). IEEE.

10. Clerget-Darpoux, F., & Elston, R. C. (2013). Will formal genetics become dispensable?. Human heredity, 76(2), 47-52.

11. Colbran, L. L., Chen, L., & Capra, J. A. (2017). Short DNA sequence patterns accurately identify broadly active human enhancers. *BMC genomics*, *18*(1), 1-11.

12. Elston, R. C., Satagopan, J. M., & Sun, S. (2012). Genetic terminology. In Statistical Human Genetics (pp. 1-9). Humana Press.

13. Erill, I., & O'Neill, M. C. (2009). A reexamination of information theory-based methods for DNA-binding site identification. *BMC bioinformatics*, *10*(1), 1-22.

14. Gibson, G. (2012). Rare and common variants: twenty arguments. Nature Reviews Genetics, 13(2), 135-145.

15. Hardiman, O., Al-Chalabi, A., Chio, A., Corr, E. M., Logroscino, G., Robberecht, W., ... & Van Den Berg, L. H. (2017). Amyotrophic lateral sclerosis. Nature Reviews Disease Primers, 3(1), 1-19.

16. Kiernan, M. C., Vucic, S., Cheah, B. C., Turner, M. R., Eisen, A., Hardiman, O., ... & Zoing, M. C. (2011). Amyotrophic lateral sclerosis. The lancet, 377(9769), 942-955.

17. Lanchantin, J. (2017). *Deep Motif: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks* (Doctoral dissertation, University of Virginia).

18. Latchman, David S. "Transcription factors: an overview." The international journal of biochemistry & cell biology 29.12 (1997): 1305-1312.

19. Lawrence, M., Huber, W., Pages, H., Aboyoun, P., Carlson, M., Gentleman, R., ... & Carey, V. J. (2013). Software for computing and annotating genomic ranges. *PLoS Comput Biol*, *9*(8), e1003118.

20. Leavitt, S. (2004). *Deciphering the genetic code: Marshall Nirenberg*. Office of NIH History.

21. Lehmann, E. L., & Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

22. Liou, C. Y., Tseng, S. H., Cheng, W. C., & Tsai, H. Y. (2013). Structural complexity of DNA sequence. *Computational and mathematical methods in medicine*, *2013*.

23. Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., ... & Moore, H. F. (2013). The genotype-tissue expression (GTEx) project. *Nature genetics*, *45*(6), 580-585.

24. Luo, X., Tu, X., Ding, Y., Gao, G., & Deng, M. (2020). Expectation pooling: an effective and interpretable pooling method for predicting DNA–protein binding. *Bioinformatics*, *36*(5), 1405-1412.

25. Manu Setty and Christina S Leslie. Seqgl identifies context-dependent binding signalsin genome-wide regulatory element maps. PLoS computational biology, 11(5):e1004271, 2015.

26. Pansarasa, O., Bordoni, M., Drufuca, L., Diamanti, L., Sproviero, D., Trotti, R., ... & Cereda, C. (2018). Lymphoblastoid cell lines as a model to understand amyotrophic lateral sclerosis disease mechanisms. *Disease models & mechanisms*, *11*(3).

27. Pavesi, G., Mauri, G., & Pesole, G. (2004). In silico representation and discovery of transcription factor binding sites. *Briefings in bioinformatics*, *5*(3), 217-236.

28. Patel, P. N., Gorham, J. M., Ito, K., & Seidman, C. E. (2018). In vivo and In vitro methods to identify DNA sequence variants that alter RNA Splicing. *Current protocols in human genetics*, *97*(1), e60.

29. Project MinE ALS Sequencing Consortium. (2018). Project MinE: study design and pilot analyses of a large-scale whole-genome sequencing study in amyotrophic lateral sclerosis. *European Journal of Human Genetics*, *26*(10), 1537.

30. Quang, D., & Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic acids research*, *44*(11), e107-e107.

31. Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J., & Kircher, M. (2019). CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic acids research*, *47*(D1), D886-D894.

32. Salekin, S., Zhang, J. M., & Huang, Y. (2017, February). A deep learning model for predicting transcription factor binding location at single nucleotide resolution. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)* (pp. 57-60). IEEE.

33. Schneider, T. D., Stormo, G. D., Gold, L., & Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. Journal of molecular biology, 188(3), 415-431.

34. Schneider, T. D. (2002). Consensus sequence zen. Applied bioinformatics, 1(3), 111.

35. Song, M., Yang, X., Ren, X., Maliskova, L., Li, B., Jones, I. R., ... & Shen, Y. (2019). Mapping cis-regulatory chromatin contacts in neural cells links neuropsychiatric disorder risk variants to target genes. *Nature genetics*, *51*(8), 1252-1262.

36. Trifonov, E. N. (1990). In Sarma, RH and Sarma, MH (eds), Structure and Methods, Vol. 1, Human Genome Initiative and DNA Recombination.

37. Wang, M. H., Cordell, H. J., & Van Steen, K. (2019, April). Statistical methods for genome-wide association studies. In Seminars in cancer biology (Vol. 55, pp. 53-60). Academic Press.

38. Wang, T. Q., & Xu, Y. (2016, January). Analysis of Effect of the Position on Weighted Degree Kernel for Splice Site Prediction. In *The International Conference on Biological Sciences and Technology*. Atlantis Press.

39. Wheeler, D. A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., ... & Rothberg, J. M. (2008). The complete genome of an individual by massively parallel DNA sequencing. *nature*, *452*(7189), 872-876.

40. Yue, T., & Wang, H. (2018). Deep learning for genomics: A concise overview. arXiv preprint arXiv:1802.00810

41. Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., & Shu, W. (2017). BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics*, *33*(13), 1930-1936.

42. Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., & Telenti, A. (2019). A primer on deep learning in genomics. *Nature genetics*, *51*(1), 12-18.

43. Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, *12*(10), 931-934