Finding Novel Pseudogenes using Machine Learning

Haley Granger and Joseph Tierno

***Abstract*—Pseudogenes have long been thought to lack a function, but have recently been found to play an important role in regulation and prevention of disease [1]. Using DanQ and ExPecto, we have shown that machine learning can be a useful tool in detecting mutations which can regulate expression in both genes and pseudogenes based on small mutations. Both methods used DeepSEA sample datasets [2] and can predict different expression effects for genomic variants. [3]. The prediction frameworks differed and the models can use this expression information for genomic sequencing data to identifying pseudogenes if associated with a loss of expression in disease related mutations.**

***Index Terms*—CNN, RNN, ExPecto, DanQ, DNA, pseudogene, genomics**

I. Introduction

When the human genome began in 1990 [4] it had hopes to solve all of the mysteries surrounding our DNA. Unfortunately, this is still not the case today and the data from the project has taught us that the situation is much more complicated than originally thought. A typical human genome is about 3 billion base pairs long [4], but since we are diploid the actual number of genomic DNA per cell is closer to 6 billion base pairs. Sequencing this much DNA reliably has proven to time consuming, and for the most part, a waste of time. This is especially the case since only about 1-2 percent of the genome [4] is responsible for coding proteins. The remaining genome has long been considered "junk" DNA because of this, but this 98 percent may have an important regulatory function which could be a target for a new field of therapeutics.

Due to the nature of natural selection and random mutations, the DNA that is present in our genome can be inferred to have some importance [5]. Previous theories suggest that most of the genome was not important, and this flexibility allowed evolution to occur [6]. The flexibility may be an important evolutionary mechanism, but there is strong evidence for a regulatory function of pseudogenes which are typically thought important in this process [7]. The fashion in which pseudogenes are capable of regulating expression is complex and this area of research could not be possible without machine learning.

Pseudogenes are defined as being an copy of a functional gene that no longer functions and have been previous called "junk DNA" [8], but this definition does not really cover everything. There are many ways to get including premature stop codons, frameshifts, or retrotransposition [8]. The mechanisms in which these happen can be simple or very complex due to the nature of the change, but in most cases seem well preserved. This would suggest that there is some evolutionary pressure to keeping these changes intact [9].

Work from the Pandolfi lab [7] found that the pseudogene for *PTEN (PTENP1)* was able to regulate the level of *PTEN*. Furthermore, they showed that *PTENP1* is lost in some cases of human cancer [1]. Together this strongly suggested that *PTENP1* played an essential role despite it being a pseudogene. The paradigm they constructed involved miRNAs competing for RNA transcripts from both *PTEN* and *PTENP1* [7]. This idea gave rise to a novel type of RNA called competing endogenous RNA (ceRNA) [10] which existed as part of the normal regulatory framework.

Future work by the same group [11] found that there were several different ways in which this regulation could occur. First, the function gene and pseudogene could be complementary and bind which would inactive the transcript, resulting in gene silencing. Second, the pseudogene could result in the creation of small interfering RNA (siRNA) which also causes gene silencing. Third, the pseudogene would act as ceRNA as previously described. Additionally, pseudogenes could compete for transcriptional machinery at the DNA level [12]. The functionality of these pseudogenes were found to be implicated in a number of diseases, if the pseudogenes were nonfunctional there was a higher chance of disease even if there was no mutation within the gene itself [11].

A majority of these mutations were due to single-nucleotide polymorphisms (SNPs) [13] which are small mutations that substitute a single base pair in DNA. These areas are commonly found in both genes and noncoding regions of DNA [13]. Aside from pseudogenes, these areas of noncoding have been implicated in occurring in areas where histones or transcription factors bind [14]. SNPs have long been known to cause disease in genes [15], but finding SNPs in noncoding regions of DNA and associating them to disease has been relatively difficult. ML has proven to be useful in finding SNPs that occur in noncoding regions via either blocking histone binding or transcription factors [3].

Histone proteins are responsible for the packaging DNA and regulate gene expression via packaging [16]. Any modifications that would alter this packaging could cause an unwanted effect during gene expression. Similarly, transcription factors are important for allowing genes to be expressed and mutations that impact this may be deleterious [17]. Finding SNPs that have impacted either of these is the mechanism of ExPecto [3] which focuses on finding SNPs and predicting if either

H. Granger and J. Tierno are with the College of Computing and Software Engineering, Kennesaw State University, Marietta, GA, 30060 USA. Emails: agrange5@students.kennesaw.edu and jtierno@students.kennesaw.edu

histones or transcription factors binding may be effected. This approach is very important in finding new SNPs which may be implicated in disease [14], [3].

Genome-wide association studies (GWASs) are commonly used in the field to find genes or mutations which play a role in disease [18]. These datasets are typically very large and complex since they are pooling information the entire genome [18]. These analyses have historically used statistics and verified results with biological assays [18] but more recently have shifted to ML [19]. Because of the randomness of ML, the success of these models are defined by their datasets, training algorithms, and architecture [19].

Here we compare two ML models which predict gene function from sequence data DanQ and ExPecto. First, we will review the history of their use, development, and how they have impacted the genomic community. Next, we compare their architecture and how they accomplish similar goals with their designs. Finally, we will explore possible future research and implications from these models and alterations to their architectures for pseudogene identification.

A. Research problem

A number of untreatable diseases have been declared "undruggable" because the pathways involved with them are so important to healthy bodily function that off target effects cannot be tolerated [20]. Finding ways to cure this diseases is important and with the advent of messenger RNA (mRNA) vaccines being widely used it may give rise to a new class of drugs. Because a vast amount of the human genome is considered to be noncoding, it is important to better understand how these regions regulate transcription of genes [21], [22]. The size of this region of the genome makes this analysis of these areas impossible without the use of machine learning.

B. Purpose of the study

The goal of this study is to detect novel pseudogenes in the human genome which may be related to diseases. It has been well established [11] that gene regulation can be heavily influenced by noncoding areas of the genome, particular pseudogenes. Finding these areas have been proven to be difficult due to the vast size of the genome and the typical human variance between genomes. Here we propose a method of using a recurrent neural network (RNN) with convolutional elements to accurately detect the possible patterns found within a genome.

C. Audience

The intended audience of this paper could be either individuals doing research in pharmacology/biochemistry or machine learning. This information could easily help individuals in the drug discovery areas by proposing new targets and providing information for complex diseases such as cancer. Individuals interested in machine learning may also benefit from this study based on the efficiency of the RNN since it is a hybrid between a RNN and a convolutional neural network (CNN). Hopefully, utlizing CNNs with RNN elements may be useful for other applications in the field with minimal modifications [23].

D. Contribution

The application of this neural network (NN) is on a scale that has yet to be accomplished. Previous papers have used this network for similar applications, but the object and design has not be as consequential [24]. Our present research intends to utilize a large amount of data to find new targets for potential genetic regulation that has not been found prior.

E. Motivation

The human genome has a large portion of it still little understood. These areas play a large role in regulation of gene expression and could be more important in disease than previously believed. If areas of a noncoding DNA could be found to be leading causes of disease, treatments could be directed to solving these issues which may be more effective than current treatments and have fewer side effects.

F. Paper goals and organization

The purpose of this paper is to show that this NN is able to analyze large amounts of DNA and find similar sequence in different parts of a individual's genome. These similarities will also need to be found across several different individuals due to the genetic variance of individuals. The paper is organized into sections which focus on different aspects of this work. Section II will introduce the science that lead to the motivation and purpose for the study. It will include limitations and possible complications from this type of research. Section III will focus on introducing the NN used in this study and the applications it has shown to be successful with. Section IV will cover the details of utilizing our NN and our dataset. Section V presents our results from analyzing the genomes with this NN and show its performance. Finally, Section VI presents our conclusion and possible next steps for this research.

II. Related works

With the genome being so large it has been difficult to map all possible interactions [14], [25]. Mapping these interactions has been a goal of bioinformatics for years [26]. With advances to machine learning (ML) this processes like this have now become possible [27]. Genomic data consists of four different letters representing base pairs. The sequence of this data is highly conserved where comparing individuals and finding patterns can can be difficult because the common motifs may be thousands of characters apart from each other [19]. Because of the repetitive nature of DNA with such a small alphabet the patterns that can exist has proven very difficult in the past.

Common architectures used for analyzing genomic data are CNNs and RNNs [19]. Each architecture has its own benefits for analyzing a particular type of data and it is important to understand the strengths of the architecture and how it will work with the dataset [19]. By analyzing these strengths it was important for the decision of why a particular model is useful for this analysis. Although CNNs and RNNs are capable at analyzing genomic data, the structure of the hiddens layers is different and connectivity [19].

CNNs architecture is composed of convolution units which take in data and break the input into feature maps [19]. These nodes are fully connected to previous layers which can be repeated or reduced in pooling layers which determine proximity of these nodes and thus relatedness between input data [19].The architecture focuses on connectivity, transition, and location which are emphasized in their convolutional and pooling layers [28].

RNNs architecture function with connections between nodes along a temporal sequence which create a temporal memory [19]. Using this memory, RNNs are better suited at identifying patterns observed in previously scene data [19]. This can be implemented to create a long short-term memory (LSTM) which addresses the long-term dependency issues associated with RNNs [19]. The LSTM node build as data is input into the RNN while the input gates determine which information is recorded in these nodes [19].

Both DanQ and ExPecto utilize CNNs in their models due to the convolutionals ability to break down these problems into a large number of features [24], [3]. Deep learning-based sequence analyzer (DeepSEA), a well established standard in standard in predicting the function of DNA from sequences, also uses a CNN for the same reason [2], [14]. The ability to break down repetitive motifs into these features has proven very useful for ML in terms of finding how these sequences relate to the genome [3]. These convolution layers are able to focus on regulatory motifs which can be found commonly used in most genes [3], [24] . In addition to using a CNN, DanQ also utilizes parts of an RNN which allow include the structure of a LSTM providing for more focus on small repeated patterns from input data [24].

DeepSEA uses a CNN to predict genomic variation based on DNA sequences by finding variations between an input sequence and a reference genome [14]. The data is then fed into the convolutional layers which was trained based on data from ENCODE, Roadmap Epigenomics, and chromatin profiles which was broken up into common motifs [14]. This allows for the prediction of how individual SNPs could change the binding of transcription factors or histones from the reference consensus sequence [14]. This prediction has proven to be a useful tool in detecting SNPs that interfere with gene expression and have been associated with disease [14].

The structure of DeepSEA was expanded upon to make ExPecto which also uses utlizes a CNN too makes its predictions [3]. ExPecto was created to compare genome information from resources such as ENCODE and ENSEMBL to determine the effects of SNPs on the function of genes using common motifs which were determined by ML [3]. Both ENCODE and ENSEMBL contain all known genome information for various organisms (including human) and compile metadata from many different projects from different labs. One of the primary datasets comes from the 1000 Genome Project which utilizes a large sample size of genomes in order to create a consensus sequence which provides a reference genome to compare an individual sequence to. This reference has changed over time and has been updated as these changes have been made.

DanQ uses a hybrid CNN/RNN to determine the effects of individual sequences using data from ENCODE as a reference sequence [24]. The predicted function can determine if the sequence is most likely to be a common motif found in genes. This can infer that a SNP may have a very different function to the typical gene which does not contain the mutation [24]. These results have given scientists a starting point to help guide their efforts in GWAS studies [24].

III. Techniques used in the study

This study implements a hybrid network, DanQ [24] and another neural network, ExPecto which is "a sequence-based expression prediction framework [3]". Both of these algorithms maintain different approaches to genomic based prediciton. DanQ is a hybrid recurrent neural network(RNN) with a bidirectional convolutional layer. ExPecto contains three main sections to its framework: a deep convolutional neural network(CNN), a spatial feature transformation module, and a linear model as seen in figure 3. This section will present an in depth review of the frameworks of each network and explain why these frameworks are important when working with genomic sequence predictions and other sequence based predictions. Later in this section we will explore the data used from the DeepSEA [2] project within each framework and focus on how we analysed and made conclusions with this data.

A. DanQ Architecture and Framework

DanQ is a hybrid neural network meaning that it implements several different types of networks layers and does not consist of just one method. The framework for DanQ can be visualized in figure 4. DanQ creators describe it as a "hybrid convolutional and recurrent neural network for predicting the function of DNA *de novo* from sequence." [24] The word *de novo* is latin for "brand new" and represents brand new functions of DNA from sequence. [24] The original implementation of DanQ is used in this study but an alternate will be explored for implementing the network to detect pseudogenes in the future.

For the original implementation, DanQ accepts large data sets of genomic sequences, from the DeepSEA [2] database, as input and runs them through a bidirectional recurrent neural network (RNN) with an added convolutional layer. A recurrent neural network is a neural network that can identify context based information. Unlike RNNs, convolutional neural networks (CNNs) are networks that specialize in identifying instances of an object or data and processing spatial information, for example: image identification and classification. [29] RNNs are able to store information before and after particular sequences and make context based connections, for example: human speech recognition. [29] This is an important architecture when looking at genomic sequences where the order of each protein matters and must be matched exactly.

For the purpose of future research regarding pseudogene identification and prediction, the implementation could be

better altered to create a unidirectional RNN. The reason the implementation should be changed is because a bidirectional RNN is not needed for identifying the start codon in the genomic sequence and it would add a lot of extra computational time. [30] This is an area and implementation that can be explored in more detail in future research. The use of a unidirectional RNN would be a valuable study for pseudogene detection. Bidirectional layers are valuable for both forward and reverse context based information such as speech recognition but in genomic sequences, the recognition of a feed forward network is sufficient. [30] Luckily, implementing either type of DanQ model are easy implementations to alter and compare the results in the future.

Before diving into the network architecture, the data must be analysed and understood. A large amount of data preprocessing must occur before running it through the neural network. The data set used is from DeepSEA which is a "deep learning-based algorithmic framework for predicting the chromatin effects of sequence alterations with single nucleotide sensitivity." [2] Along with the framework for predicting chromatin effects, DeepSEA has an open access database that has preprocessed data for portions of the human and other organisms genomes. The database hold three types: co-expression, GSEA microRNA targets, interactions, and TF binding data which can be used in different types of research. [2] DanQ recommends using the training bundle and this is the data that our research used. It is a zipped .tar file that contains training, testing, and validation .mat data files from DeepSEA. All of these are preprocessed and labeled so they can be directly fed into the network. Unlike other implementations, the data is loaded using h5py and scipy which are libraries that allow access and readability of .mat files.

After these files are opened in the script, they are then cast into NumPy arrays. [31] DanQ uses the transpose() and array() methods to first create the axis and dimensions of the input array and then insert the data into a NumPy array [31]. NumPy structures the array and makes the data easier to evaluate and feed into the RNNs layers. After the data preprocessing, the network is created with multiple python packages which will be discussed below.

The main software packages that are used to run DanQ are Keras [32], Tensorflow [32], Theano [33], NumPy [31], and seya [34]. Keras is built on the Tensorflow platform and is used to create the RNN input layers, hidden layers, bidirectional layers, long-short term memory cells (LSTM), and output layer. Tensorflow and Keras are machine learning platforms which allow users to efficiently execute "low-level tensor operations on CPU, GPU, or TPU" [32], compute gradients of "arbitrary differentiable expressions" [32], scale computation, and export external data like graphs to other sources [32]. There are two core data structures that compose Keras: layers and models. A sequential model, which DanQ implements, is a linear piling of layers. Keras provides other functions that allow for more complex creation of layer stacking and architectures as well which can be a possibilty for future implementations of a similar structured network.

Within the sequential model created with Keras, DanQ adds a one dimensional convolutional layer with a rectified linear unit (ReLu) activation function, two sequential layers, and an long-short term memory (LSTM) layer [24]. The ReLu activation function uses the formula:

$$f(x) = max(0, x)$$

and simply returns a 0 if any negative input is received and returns the value it received back if it is positive. This creates "an output that has a range from 0 to infinity [35]." This convolutional layer makes DanQ a hybrid recurrent neural network and adds another set of input layers to the implementation. Convolutional layers allow a feature map to be created of the data and increase prediction accuracy of the RNN. This is an important part of the implementation and adds another function in prediction. Future research could be done on the impact of having or removing the convolutional layer and the networks affected prediction accuracy.

After the convolutional layer, the bidirectional RNN is implemented, then two sequential layers are created. The first sequential layer has input dimensions of 75x640 with an output dimension of 925 and the second has input dimensions of 925 with an output dimension of 919. [24] Aside from the dimensions of the layers being different, the first layer assigns ReLu as the activation function and the second has a logistic sigmoid function as an activation function. A sigmoid function maps a line with a small range(0,1) and can the be interpreted as a probability. A logistic sigmoid function can be expressed in the formula:

$$S(x) = \frac{e^x}{(e^x) + 1)}$$

There are differences between the two functions and advantages to both. The ReLu function allows for faster calculations and eliminates the vanishing gradient problem [36]. The advantage of having a layer with a sigmoid function is that this function allows for easier prediction of a start codon within a genomic sequence. The prediction is binary, the start codon is either in the sequence or it is not, and the use of the sigmoid function allows for a logistic regression prediction and makes binary prediction better. [36] The use of both of these functions will allow the layers to eliminate some of the vanishing gradient problem found with the sigmoid function as well as predict more binary results that the ReLu function can not offer.

Another important aspect of the DanQ implementation is that it incorporates two LSTM cells. RNNs are context based neural networks which loop through the layers in a cyclical fashion [37].A LSTM cell has four layers that each have unique functions unlike a traditional RNN layer which has one. Normally, with smaller data sets, RNNs have no problem predicting future occurrences from previous context. The issue occurs when there is a large data set and it is difficult for the RNN to predict from such a large number and make accurate connections between data. This problem is known as the vanishing gradient problem, a problem which LSTM cells are the solution [38].

There are several reasons this study uses DanQ as an implementation method. First, the use of different activations functions allows for better prediction of sequence variations. Future research can allow for this implementation to better predict PTEN pseudogenes as well. Second, the implementation of the LSTM cells increases the accuracy of predictions with a large dataset such as a genomic sequence. Finally, the implementation of DanQ has been used in similar research and is cited as an accurate source for sequence prediction in the genomics field. [24]

B. ExPecto Architecture and Framework

While ExPecto and DanQ both have similar goals of genomic sequence predictions, they have different approaches. ExPecto creators describe their framework as helpful for "predicting expression effects of human genome variants *ab initio* from sequence...and training new sequence-based expression model with any expression profile" [3] The term *ab initio* is latin for "at the beginning" indicating that ExPecto starts from the beginning of the sequence. This definition of ExPecto means it focuses on predicting mutations and disease risks within genes. The framework of ExPecto can be visualized in figure 3. The main difference in the framework is that DanQ uses the Keras package which creates already built convolutional layers and neural network layers. [32] This makes for a simple implementation but does not allow for specific data preprocessing to occur depending on the sequence being fed through the network. ExPecto uses a parser to create custom built arguments on how, when, and what can be fed through the network. [3]

The main libraries and packages that this framework uses are: NumPy [31], h5py [39], Pandas [40], SciPy [41], Six [42], XGBoost [43], and PyTorch [44]. NumPy allows users to convert data into arrays and structures that can be fed into a network while h5py works in conjunction with NumPy to translate that data into readable models for further prediction. Other libraries worth elaborating on are XGBoost which "implements machine learning algorithms under the Gradient Boosting framework" [43] and the DeepSEA database which is not a library but an algorithmic framework that allows ExPecto to be pretrained on "predicting chromatic effects of sequence alterations with single nucleotide sensitivity." [2]

The first part of the ExPecto framework is a deep convolutional neural network (CNN) the purpose of this layer of the framework is to "transform genomic sequences to epigenomic features." [3] This makes it so that the genomic data is easier to train and be processed by the network. DeepSEA works with ExPecto to predict the epigentic states from sequences that are fed into the convolutional layers of ExPecto. The second part of the ExPecto framework is the spatial transformation functions that create feature maps from the output of the convolutional network. [3] This is similar to the convolutional layer of DanQ where both create feature maps for better prediction accuracy. This transformation also decreases the size of the data to a manageable level and compresses it to feed through the rest of the network. Weights are updated during this stage of the network and a comparison of one weights document in the sample data file from ExPecto can be seen represented in figure 5.

The last part of the framework makes "tissue-specific expression predictions" [3] which are used to predict "gene expression levels for each tissue" [3]. The architecture used linear regression models in tandem with the gradient boosting algorithm found with the XGBoost library. [43]

C. Data Analysis and Methods

In the previous sections describing both the DanQ and ExPecto frameworks, the datasets were mentioned and described to give a full picture of how they were fed into the architecture of each network. This section will provide a summary of the datasets and how we analysed our data after feeding it through each network.

DanQ and ExPecto both use DeepSEA for their database but in different ways. [2] We used the DeepSEA training files for DanQ which are split between training, validation, and testing files. These three are fed through DanQ and are formatted in .mat file types. They are

ExPecto uses several different data files for it's numerous applications. We used the sample data file that comes in a .csv format. The data comes from the Ensembl [45] which is a genomic annotation database that is great for feeding preprocessed data into networks.

The csv file has headings of: id, symbol, seqnames, strand, TSS, CAGE representative TSS, and type. Each of these allow for ExPecto to sort the data. The id represents the ensembl gene id which is specific to each gene. Ensembl [45] is one of the databases that ExPecto uses as a reference to their data and they took their sample data from it. The next variable in the data set is segnames which represents the segment name or the chromosome the gene resides on, strand is represented as a (+) or (-) sign and shows which side of the DNA the strand is on. Transcription start sites or TSS refer to where the expression starts within the gene and where it will start making RNA. If there is a (-) sign then it annotates a reverse strand whereas a (+) sign refers to a forward strand. The variable cage rep indicates where the expression starts or ends, depending on the plus or minus sign indication in the TSS variable. Type shows what the gene does and the main two types seen in the sample data are protein coding and lincRNA(long intergenic non-coding RNA). [46] LincRNA refers to the proteins that do no code which indicates they could be a pseudogene. Some can be SNPS(single nucleotide polymorphism) where they refer to a single point in the sequence that can cause different alterations and effects. The first entry in the sample set is from the genome: GRCh38.p13. [45] These variable are all important in training the ExPecto network and future research can be done using ExPecto and prediction of pseudogenes due to how this data is organized. last base pair that it is using

haplotype

DanQ and ExPecto both have sample results from pretrained models that we found to be beneficial in our results and methods of analysis for this research. DanQ generated motifs which

are word clouds for base pairs in genes. Our interpretation of these motifs and their offsets, p-values, e-values, and q-values are represented in figure 1.

IV. Proposed Method

Our dataset will be a subset of the NCBI 1000 Genome Project [47]. Their dataset provides thorough data with a wide range representing geographical variation while minimizing gender biases. The eventual goal would be to use as many of these samples as possible for mapping the whole genome, however not every individual has their entire genome sequences. Additionally the files themselves are very large, with a typical file being 15 GBs. Because of this, initial training will focus on *PTEN* first which is substantially smaller and easier to validate. With the dataset being relatively large compared to similar studies and having representatives of very diverse geographical populations the patterns extracted should represent common trends for humans as a whole.

Utilizing the different file types required for DanQ has proven to be a challenge with this dataset. The data typically comes in bam, sam, or fasta files which are commonly used in genomics. DanQ does not readily take these files, nor many other types of NNs or software. So the files need to be converted before being input into the model. Although this may take a significant amount of time, this does allow for a more supervised approach at training with the input data.

Using DanQ as our model for this study, we will be taking full advantage of both types of layers in the NN [24]. After attempting to use a fully recurrent system, it quickly determined that the amount of nodes for the LSTM may not be practical. *PTEN* encompasses a little over 108,000 base pairs (source for length of *PTEN*). In an effort to reduce the number of nodes, the sequence is converted to its amino acid sequence using the sense strand. This reduces the input string length by 3, while increasing the alphabet to 21. The other added benefit to this is that converting the gene sequence will allow for some person to person variance to be minimized due to small point mutations in the gene. Ultimately this should allow for a better prediction of consensus sequences when analyzing different gene from individuals.

V. Results and Discussions

Our findings of this study can be broken down between Results and Discussion sections and the subsections are labelled accordingly. Our results and furthermore discussion can be broken down into the following sequential categories: results of DanQ, results of ExPecto, results of the RNN we created, results of the datasets used. Each section will be formatted by iterating through these topics and identifying new results found from the data sets we have previously described.

A. Results

First, we found that the motif files from DanQ which were created from the trained model have several variables including: optimal offset, p-value, E-value, q-value, and overlap. These are all important in analysing the results from the DanQ

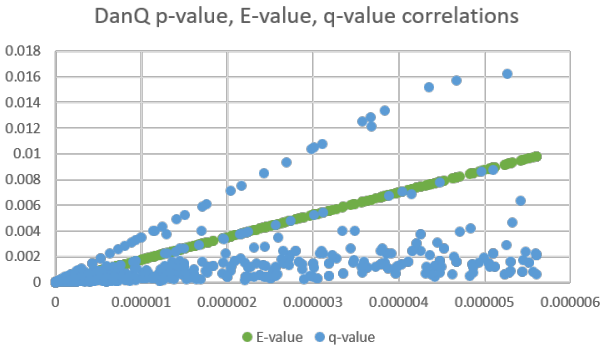

Fig. 1. Plotted points of p, q, and E-value correlations for the DanQ model [24]

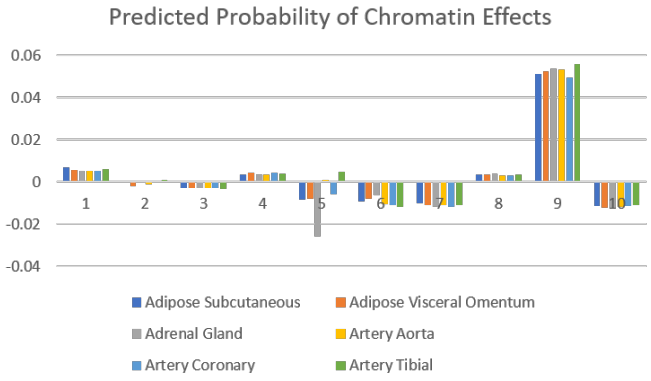

Fig. 2. Output from trained expecto model of probability of chromatin effects [3]

model. Figure 1 shows the correlation between the p-value, E-value, and q-values of all the motifs generated from the sample DanQ model. The green linear line shows the E-value plotted with the p-value along a scale from 0 to 0.000006. The blue more scatter points represent the q-value plotted with the p-value along the same scale.

The time it takes to train DanQ is relatively long. It is estimated that it takes 6 hours per epoch with a GPU system running. Because of the large amount of data, the computational time is relatively slow. In order to get faster computational time while maintaining accuracy of the loss function, we used the binary cross entropy loss function to allow for binary loss optimization. The higher this accuracy is, the better our model will be able to identify *PTEN* over other genes.

Next, we found the results of ExPecto. ExPecto's output from their chromatin prediction script shows the chromosome, the start sequence, the gene it resided on, strand, and percentage likelihood of the sample having certain chromatin effects or abnormalities. Figure 2 shows a sample of the output data from a trained ExPecto model using it's chromatin effect prediction script.

In regards to other observations made through this research, another valuable thing to note is that samples do not have a

lot of variance between genomes. Most genes have a high degree of conservation when looking at a large population (1000 genome project). This information will also be discussed below in the discussion section.

B. Discussion

The first results we will discuss are those from the DanQ model. We created figure 1 which shows the p-value correlations plotted with the q and E-values. There is a linear correlation between p and E-values which means that the probability versus expected values correlate strongly and show that DanQ has accurate predictions with its pretrained model. There are less correlations between the q and p values which means that the false positive discovery rate does not necessarily correlate with false positives. This is a good sign showing that the network is trained to identify some false positives but it could use some improvement and alterations to catch hard to identify pseudogenes or false positives within the sequences.

When we stated the time it takes for DanQ to be trained, it would be important to optimize the loss function as quickly as possible. This loss function could be used for all subsequent inputs because it would not be practical to optimize a loss function using a full genome. This ultimately could lead to a bias towards detecting *PTEN* over other novel genes. This is our prediction for *PTEN* but table I

TABLE I
PTEN-201 HAPLOTYPES OUT OF 6000 POPULATION

Haplotype	Frequency(count)
REF	0.999(6009)
268D¿E	0.000997(6)
19D¿E	0.000166(1)
10S¿G	0.000166(1)
289K¿E	0.000166(1)

Figure 2 shows a sample of the output data from a trained ExPecto model using it's chromatin effect prediction script. This shows that ExPecto was able to predict the likelihood of different chromosomes holding different mutations within the gene. There should be no correlation and the likelihood should increase if it is accurately predicted to be a mutation.

As we said in the results section above, samples do not have a lot of variance between genomes. Most genes have a high degree of conservation when looking at a large population (1000 genome project). Through investigation and identification we concluded that this dataset primarily contains individuals which are phenotypically healthy and thus SNPs which are related to disease may not be properly represented here. This is an important observation since models currently do not know how to account for these subtle shift or alterations. Further research and implementation of our RNN could help to predict subtle changes where other models can not.

After more training and alterations, our model could be able to successfully detect *PTEN* when given random sequences of DNA. We would use indicators of the precision and recall and find X and Y respectively. These values were could lead to our F_1 score of Z, which would indicate if our data is valid or

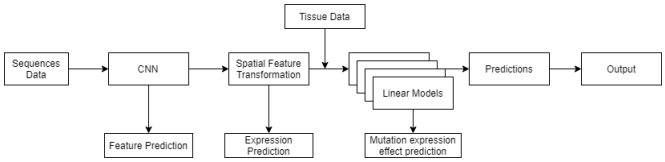

Fig. 3. ExPecto Framework [14]

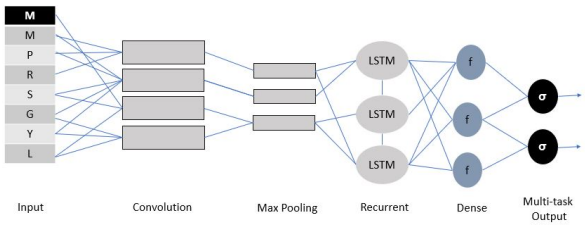

Fig. 4. A schematic of the DanQ architecture [24]

invalid. False positives need to be taken into consideration and will be generated by our method, which would suggest whether our method is good or bad. Overall, the accuracy of prediction with our model can improve with each iteration and by taking both frameworks of ExPecto and DanQ, future research in the area of identifying and predicting *PTEN* is crucial in understanding more about the RNN and the corresponding prediction variables.

We found that there was not much variation in the data inside a specific gene and in order to conclude further about most haplotypes for PTEN-201, future research could be done focusing on specific genes.

but here is a sample of PTEN-201: ref (0.99) 268D¿E from d to e (found in 6 people), point mutation only happened at one point, changed the amino acid from one to another 6/6000 19D¿E 1 person 10S¿G deleterious(not ideal and caused a large change that could be serious) 1 person 289K¿E 1 person

Another thing we found is that the data does not see all the shifts between particular amino acids, if there is no change from letter to letter but there is still a shift, its not reported in the haplotype this means a genome with new SNPs that is not reported or detected is still undiscovered in this field. If this was present in a pseudogene it could suggest loss of function which could be associated with a disease. This was not the purpose of DanQ and ExPecto but it is useful for future research endeavors using our RNN and a modified framework.

Figure 4 represents the architecture of DanQ which is written in detail in Section 3 of this paper. The input is fed into the hybrid-recurrent neural network which is then pooled similarly to traditional CNNs. Following this, it utilizes a LSTM cell which is characteristic of RNNs. After this there are two dense layers and then the output is generated. The architecture of DanQ can be represented in a linear fashion.

Our networks were trained with several data sets which are known and optimized for use with neural networks. We could find data which are known to have pseudogenes associated with diseases [11]. We would list the genes our RNN was

trained with and compares the size of each training sequence to the pseudogene we hoped to detect. Because of genetic variation between individuals, our RNN might not be able to detect every pseudogene but should be able to detect a majority of them when given a large sequence adjacent to the pseudogene. This will suggest that our model is ready to find unknown sequences which could potentially regulate the expression of these genes. This is an important observation about how our model can be altered to predict and find these unknown sequences.

Table III is adapted from [11] comparing a few know gene/pseudogene combinations which result in regulation. The typical size difference between the gene and pseudogene is easily seen here. After further modificiations, we predict that our RNN could detect pseudogenes based on training data with their functional gene.

Table II shows a comparison of the major libraries used in both ExPecto and DanQ. They share some functionalities but have different libraries to implement their networks and frameworks.

TABLE II
A COMPARISON OF LIBRARIES AND PACKAGES USED IN BOTH DANQ AND EXPECTO

Libraries	ExPecto	DanQ
Python	✓	✓
SciPy	✓	✓
PyTorch	✓	X
NumPy	✓	X
Pandas	✓	X
XGBoost	✓	X
Theano	X	✓
Keras	X	✓
Seya	X	✓

The weights and biases from ExPecto were used during the training of the model and are used in the prediction scripts as well. Figure 5 shows the bias of -0.0495619 and its weights used for prediction over time. This is a single instance of the multitude of bias data sets that ExPecto runs through to allow for more randomization of the framework.

TABLE III
COMPARING RELATIVE SIZES OF GENES TO PSEUDOGENES [11].

Gene	Length (bp)	Pseudogene	Length (bp)	Gene/Pseudo
PTEN	108,305	PTENP1	3,916	27.66
KRAS	45,683	KRASP1	5,232	8.73
BRAF	211,601	BRAFP1	3,263	64.85
CYP4Z1	62,795	CYP4Z2P	57,380	1.09

After further modification of our RNN we should be able to establish that our RNNs were able to detect pseudogenes reliably, we will begin scanning the full genome for other potential matches. By scanning the genome of a certain amount of individuals we could create a consensus sequence and find that different regions could potentially create a transcript which would interact with an unknown variable. This could suggest that this region could play a regulatory role in the proper expression of that unknown variable. Table IV shows the region of interest for different genes and additional regions of interest for other genes that would be beneficial to trained our RNN with. Our RNN and other frameworks like ExPecto and DanQ used in this study could potentially be trained on additional genes and find even more regulatory regions which could be implicated into other disease phenotypes.

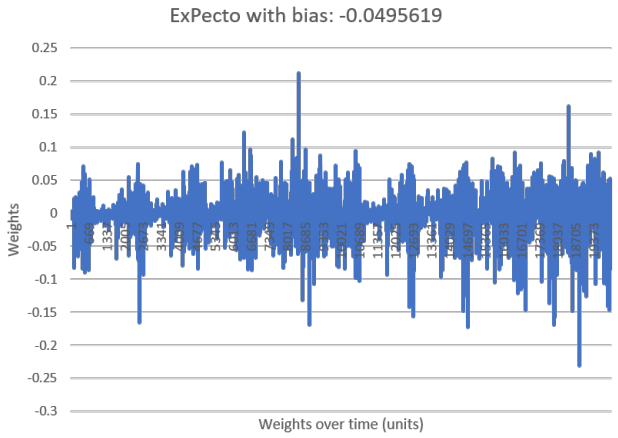

Fig. 5. ExPecto weights over a period of time in units used for prediction [3]

TABLE IV
THIS TABLE COMPARES THE DIFFERENCES BETWEEN IN LOCATION OF GENE AND PSEUDOGENES [11].

Gene	Chromosome	Pseudogene	Chromosome
PTEN	10	PTENP1	9
KRAS	12	KRASP1	6
BRAF	7	BRAFP1	X
CYP4Z2P	1	CYP4Z2P	1

VI. CONCLUSION

Both DanQ and ExPecto are able to find SNPs from sequence data using datasets from DeepSEA. These mutations can be associated with a loss of expression based on the comparing to a reference genome and determining if this SNP may impact binding to histone complexes or transcription factors. The overall outcome of this change in expression can only be inferred to be relevant in a disease using these models and until this hypothesis is tested by biochemical and GWAS studies the actual validation cannot be determined. By utilizing these models and finding they these SNPs are occur in noncoding regions of DNA it can be suggested that this is a pseudogene.

Determining if the noncoding region is a pseudogene requires comparison of sequences to known genes. This would require the noncoding regions to be analyzed with a model that compares the similarity of the sequence to all known genes within the human genome. Some smaller sequences would be difficult to associate to genes since many known pseudogenes are thousands of base pairs long (*PTENP1*, *KRASP1*, *BRAFP1*,

and *CYP4Z2P*). Furthermore, since pseudogenes are able to be found on different chromosomes from the gene (such as *PTENP1*, *KRASP1*, and *BRAFP1*) it is even more challenging to confidently claim that small regions of DNA are copies of existing genes.

Future research implementations have been discussed throughout this paper as they are necessary to continue genomic prediction particularly with pseudogenes.

VII. Acknowledgements

We would like to acknowledge Dr. Mohammed Aledhari for his support and encouragement to undertake such an ambitious project in his Machine Vision/Learning class.

References

[1] R. C. Pink, K. Wicks, D. P. Caley, E. K. Punch, L. Jacobs, and D. R. F. Carter, "Pseudogenes: pseudo-functional or key regulators in health and disease?" *Rna*, vol. 17, no. 5, pp. 792–798, 2011.

[2] J. Zhou and O. Troyanskaya, *What is DeepSEA?*

[3] J. Zhou, C. L. Theesfeld, K. Yao, K. M. Chen, A. K. Wong, and O. G. Troyanskaya, "Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk," *Nature genetics*, vol. 50, no. 8, pp. 1171–1179, 2018.

[4] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt *et al.*, "The sequence of the human genome," *science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[5] A. Telenti, C. Lippert, P.-C. Chang, and M. DePristo, "Deep learning of genomic variation and regulatory network data," *Human molecular genetics*, vol. 27, no. Supplement_R1, pp. R63–R71, 2018.

[6] T. D. Price, A. Qvarnström, and D. E. Irwin, "The role of phenotypic plasticity in driving genetic evolution," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. 1523, pp. 1433–1440, 2003.

[7] L. Poliseno, L. Salmena, J. Zhang, B. Carver, W. J. Haveman, and P. P. Pandolfi, "A coding-independent function of gene and pseudogene mrnas regulates tumour biology," *Nature*, vol. 465, no. 7301, pp. 1033–1038, 2010.

[8] Y. Tutar, "Pseudogenes," *Comparative and functional genomics*, vol. 2012, 2012.

[9] T. Miyata and H. Hayashida, "Extraordinarily high evolutionary rate of pseudogenes: evidence for the presence of selective pressure against changes between synonymous codons," *Proceedings of the National Academy of Sciences*, vol. 78, no. 9, pp. 5739–5743, 1981.

[10] L. Salmena, L. Poliseno, Y. Tay, L. Kats, and P. P. Pandolfi, "A cerna hypothesis: the rosetta stone of a hidden rna language?" *Cell*, vol. 146, no. 3, pp. 353–358, 2011.

[11] L. Poliseno, A. Marranci, and P. P. Pandolfi, "Pseudogenes in human cancer," *Frontiers in medicine*, vol. 2, p. 68, 2015.

[12] Z. Darieva, A. Clancy, R. Bulmer, E. Williams, A. Pic-Taylor, B. A. Morgan, and A. D. Sharrocks, "A competitive transcription factor binding mechanism determines the timing of late cell cycle-dependent gene expression," *Molecular cell*, vol. 38, no. 1, pp. 29–40, 2010.

[13] R. Leslie, C. J. O'Donnell, and A. D. Johnson, "Grasp: analysis of genotype–phenotype results from 1390 genome-wide association studies and corresponding open access database," *Bioinformatics*, vol. 30, no. 12, pp. i185–i194, 2014.

[14] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature methods*, vol. 12, no. 10, pp. 931–934, 2015.

[15] K. Zhang, Z. S. Qin, J. S. Liu, T. Chen, M. S. Waterman, and F. Sun, "Haplotype block partitioning and tag snp selection using genotype data and their applications to association studies," *Genome Research*, vol. 14, no. 5, pp. 908–916, 2004.

[16] J. J. Parmar and R. Padinhateeri, "Nucleosome positioning and chromatin organization," *Current Opinion in Structural Biology*, vol. 64, pp. 111–118, 2020.

[17] M. Slattery, T. Zhou, L. Yang, A. C. D. Machado, R. Gordân, and R. Rohs, "Absence of a simple code: how transcription factors read the genome," *Trends in biochemical sciences*, vol. 39, no. 9, pp. 381–399, 2014.

[18] M. Civelek and A. J. Lusis, "Systems genetics approaches to understand complex traits," *Nature Reviews Genetics*, vol. 15, no. 1, pp. 34–48, 2014.

[19] L. Koumakis, "Deep learning models in genomics; are we there yet?" *Computational and Structural Biotechnology Journal*, 2020.

[20] H. He, B. Liu, H. Luo, T. Zhang, and J. Jiang, "Big data and artificial intelligence discover novel drugs targeting proteins without 3d structure and overcome the undruggable targets," *Stroke and Vascular Neurology*, pp. svn–2019, 2020.

[21] Z. R. Chalmers, C. F. Connelly, D. Fabrizio, L. Gay, S. M. Ali, R. Ennis, A. Schrock, B. Campbell, A. Shlien, J. Chmielecki *et al.*, "Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden," *Genome medicine*, vol. 9, no. 1, p. 34, 2017.

[22] S. R. Rashkin, R. E. Graff, L. Kachuri, K. K. Thai, S. E. Alexeeff, M. A. Blatchins, T. B. Cavazos, D. A. Corley, N. C. Emami, J. D. Hoffman *et al.*, "Pan-cancer study detects genetic risk variants and shared genetic basis in two large cohorts," *Nature communications*, vol. 11, no. 1, pp. 1–14, 2020.

[23] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[24] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[25] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

[26] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2063–2079, 2018.

[27] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, and A. Telenti, "A primer on deep learning in genomics," *Nature genetics*, vol. 51, no. 1, pp. 12–18, 2019.

[28] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[29] D. I. D. Learning, *8. Recurrent Neural Networks*.

[30] ——, *9.4. Bidirectional Recurrent Neural Networks*.

[31] S. Community, *NumPy v1.19 Manual*, 2020.

[32] K. Team, *About Keras*, 2020.

[33] PyPI, *Theano*, 2020.

[34] EderSantana, *seya*, 2016.

[35] H. Mujtaba, *What is Rectified Linear Unit (ReLU)? — Introduction to ReLU Activation Function*, 2020.

[36] T. Wood, *Sigmoid Function Definition: What is the Sigmoid Function?*, 2020.

[37] A. Busia, G. E. Dahl, C. Fannjiang, D. H. Alexander, E. Dorfman, R. Poplin, C. Y. McLean, P.-C. Chang, and M. DePristo, "A deep learning approach to pattern recognition for short dna sequences," *BioRxiv*, p. 353474, 2019.

[38] M. Walia, *Long Short Term Memory (LSTM) and how to implement LSTM using Python*, 2020.

[39] PyPI, *h5py*.

[40] pandas development team, *pandas*.

[41] S. Community, *SciPy v1.5.4 Reference Guide*.

[42] PyPI, *Six: Python 2 and 3 Compatibility*.

[43] xgboost developers, *XGBoost Documentation*.

[44] pytorch developers, *PyTorch Documentation*.

[45] A. D. Yates, P. Achuthan, W. Akanni, J. Allen, J. Allen, J. Alvarez-Jarreta, M. R. Amode, I. M. Armean, A. G. Azov, R. Bennett *et al.*, "Ensembl 2020," *Nucleic acids research*, vol. 48, no. D1, pp. D682–D688, 2020.

[46] M. N. Cabili, C. Trapnell, L. Goff, M. Koziol, B. Tazon-Vega, A. Regev, and J. L. Rinn, "Integrative annotation of human large intergenic noncoding rnas reveals global properties and specific subclasses," *Genes & development*, vol. 25, no. 18, pp. 1915–1927, 2011.

[47] L. Clarke, X. Zheng-Bradley, R. Smith, E. Kulesha, C. Xiao, I. Toneva, B. Vaughan, D. Preuss, R. Leinonen, M. Shumway *et al.*, "The 1000 genomes project: data management and community access," *Nature methods*, vol. 9, no. 5, pp. 459–462, 2012.