Splice Junction Site Detection using Deep Learning

Moein Hasani[*]
moein.hasani@usask.ca
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

Linglin Jin[†]
lingling.jin@cs.usask.ca
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

ABSTRACT

This project will investigate the effectiveness of different Neural Networks in the classification of DNA sequences. The purpose is to identify sequences that include Splice Junction sites and classify the sites into 'Exon/Intron' and 'Intron/Exon' or 'Neighter' classes. The are Neural networks used in this work, and the data has been processed using the K-mer method with K values of 3, 5, and 7. The final results show that smaller K and Artificial Neural Network and Convectional Neural network can produce better results than other combinations. An accuracy of 97% is achieved in this task. My implementation of the work is available at github.com/Moeinh77/DNA-sequene-classificaiton.

KEYWORDS

Splice Junction, exon, Intron, DNA sequence, Convolutional Neural Networks, Recurrent Neural Networks

ACM Reference Format:
Moein Hasani and Linglin Jin. . Splice Junction Site Detection using Deep Learning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages.

1 INTRODUCTION

DNA sequence classification has been a topic of interest in Bioinformatics research for many years [1]. Recognizing different sequences and classifying them is a step closer to the treatment of many genetic disorders. Now that we have more computational power and modern algorithms such as Neural Networks, this task is easier and more accurate than in the past.

This work focuses on the detection of Splice Sites between Exons and Introns in a sequence. Exons are sections of the DNA that code for making a protein so they are kept in mRNA. Conversely, Introns are sections of the DNA that do not code for making a protein. Thus, Introns are considered useless during transcription. I have used the Splice Junction Dataset for my work, and I aim to classify DNA sequences based on whether or not they contain a Splice Site between Introns and Extrons.

2 MODELS

2.1 Artificial Neural Networks (ANNs)

They are a group of machine learning algorithms inspired by the structure of the brain. There are certain computational nodes in ANNs that are called neurons. Each ANN has a certain number of layers and in each layer, there are several neurons, and neurons are connected to other neurons in the layer before them and the layer after them. The ANN network utilized in this project has 2 hidden layers (not considering the output layer) with 12 and 8 neurons.

2.2 Recurrent Neural networks (RNNs)

They are a class of NNs where connections between neurons form a uni or bi-directional graph. These networks are known to have memory and they are able to process data that requires a memory of the past or needs a look into the future like sound or text data [6]. There are different variations of the RNNs such as normal RNNs, Long short-term memory (LSTM), and Gated Recurrent Unit (GRU). Since the DNA sequences can be processed as text-like data, we can apply a network such as LSTM to our dataset. The RNN network of this project is a two-layer LSTM network with a hidden unit size of 50.

2.3 Convolutional Neural Networks (CNNs)

These are a type of ANNs that use convolution operations and are mostly applied to image data. These networks are often trained much faster than usual ANNs since the weight is shared between different layers in CNNs. These networks are also applied to Natural Language Processing (NLP) tasks as well. Since the DNA classification task is similar to Text Classification, these networks can be utilized for tasks like sequence classification [4]. The CNN used in this work includes one layer of 1-dimensional convolution layer with 100 kernels of size 3 followed by a pooling layer. The result of pooling is fed to a dense layer.

3 DATA

3.1 Dataset

There are 3190 DNA sequences of length 60 in this dataset, I have used 70% of the data (2233 sequences) for training and 30% for testing (957 sequences). The data have 3 columns the first column indicates the class, the second column shows the donor of the sequence and the third is the sequence. There are 3 classes in the dataset Intron/Exon IE, Exon/Intron (EI), or neither (N), and the distribution of these classes in the data is shown in Figure 1.

3.2 Data Processing

I have converted the sequences to k-mers. In bioinformatics, k-mers are sub-strings of a sequence with a length of k. Using k-mers we convert each of the k nucleotides into a word. E.g. with k= 3 the sequence 'CCAGCTG' turns into ['cca', 'cag', 'agc', 'gct', 'ctg']. K-mer method has been used by other works such as [2] and [5] and has been proved to be effective. We make a dictionary of these k-mer

[*]
[†]

Conference'17, July 2017, Washington, DC, USA
©
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00

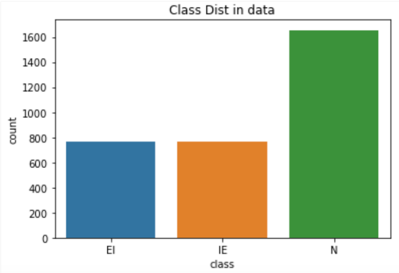

Figure 1: Distribution of the classes in the dataset

Table 1: Models Training Results

2*	**K values**					
	3		5		7	
Models	**Test Accuracy and the Training Time**					
ANN	**0.97**	25 sec	0.946	**23 sec**	0.796	**24 sec**
CNN	0.967	**21 sec**	**0.947**	34 sec	0.711	32 sec
RNN	0.921	69 sec	0.833	40 sec	**0.866**	55 sec

Table 2: How K size increases the vocabulary size (unique combination of nucleotides)

K	Vocabulary Size
3	88
5	1075
7	15351

words, and we assign a number to each word in the dictionary so when we give the sentences to the network each word is a number.

Before we feed the words (their integer equivalent) to the neural network layers, we use a technique called Word Embedding[3]. This is a method of representing words for the tasks related to text analysis. The words are transformed to spaces of vectors with arbitrary dimensions such that the words that are closer in the vector space are expected to have the same or close meanings. Word embedding can be obtained using a certain NLP technique. I used the n = 50 for embedding. This technique has been proven to increase the accuracy of the model significantly. Each k letter combination of nucleotide becomes a vector with a length of 50 and the vectors are also learned by the model.

4 RESULTS

In this section, I have provided the test results from training the three introduced Neural Networks. Each model has been trained with three different K values of 3, 5, and 7. The accuracy of each model and the time required for each model to achieve the accuracy has been stated in Table 1. The models are trained on an RTX 2060 GPU. Python programming language has been used for implementing the code and the models are created using the Tensorflow library.

As you can see in the Table 1, when K = 3 all the three models have almost similar accuracy. But as the K increases, their performance starts declining. I believe it is because bigger K values create more different combinations as shown in **??** of nucleotides and a high number of combinations can be confusing for the models I have used. The overall performance of the models I believe to be satisfactory. With K = 3 the accuracy of 97% is achieved by an ANN in 25 seconds which indicates the effectiveness of Neural Networks in the classification of DNA sequences. The heatmap for the ANN with K = 3 is shown in Fig **??** and as you can observe the model can predict the classes, despite the existence of imbalance in the dataset, pretty well.

5 CONCLUSION

In this work, I tried experimenting with different Neural Networks for the task of classifying DNA sequences. The method for reading the data (using k-mer method) showed to be quite useful and this project indicates we can treat sequence processing as an NLP task and many techniques of ML that is used for text processing can be applied to bioinformatics sequence information as well. Different values of K in k-mer method produced different outcomes and it seems that smaller values of K work better when it comes to detection of Splice Junction Site. I believe since smaller K produces fewer unique tokens, the result is more manageable for the networks. And about the networks, CNN and ANN can be considered better choices than an RNN when it comes to selecting a Neural Network for sequence classification. Since they achieve more accurate results in a more efficient matter.

REFERENCES

[1] Mikhail S Gelfand. Prediction of function in dna sequence analysis. *Journal of Computational Biology*, 2(1):87–115, 1995.

[2] MASANORI Higashihara, JOVAN DAVID Rebolledo-Mendez, YOICHI Yamada, and KENJI Satou. Application of a feature selection method to nucleosome data: accuracy improvement and comparison with other methods. *WSEAS Transactions on Biology and Biomedicine*, 5(5):95–104, 2008.

[3] Daniel Jurafsky and James H Martin. Speech and language processing (draft). *preparation [cited 2020 June 1] Available from: https://web. stanford. edu/˜ jurafsky/slp3*, 2018.

[4] Feriel Ben Nasr and Afef Elloumi Oueslati. Cnn for human exons and introns classification. In *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 249–254. IEEE, 2021.

[5] Ngoc G Nguyen, Vu Anh Tran, Dau Phan, Favorisen R Lumbanraja, Mohammad Reza Faisal, Bahriddin Abapihi, Mamoru Kubo, and Kenji Satou. Dna sequence classification by convolutional neural network. *Journal Biomedical Science and Engineering*, 9(5):280–286, 2016.

[6] Aditi Sakalle, Pradeep Tomar, Harshit Bhardwaj, Divya Acharya, and Arpit Bhardwaj. A lstm based deep learning network for recognizing emotions using wireless brainwave driven system. *Expert Systems with Applications*, 173:114516, 2021.

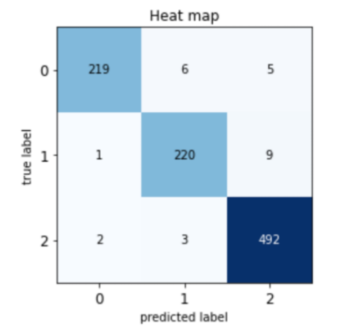

Figure 2: Heatmap of predictions with ANN with K = 3