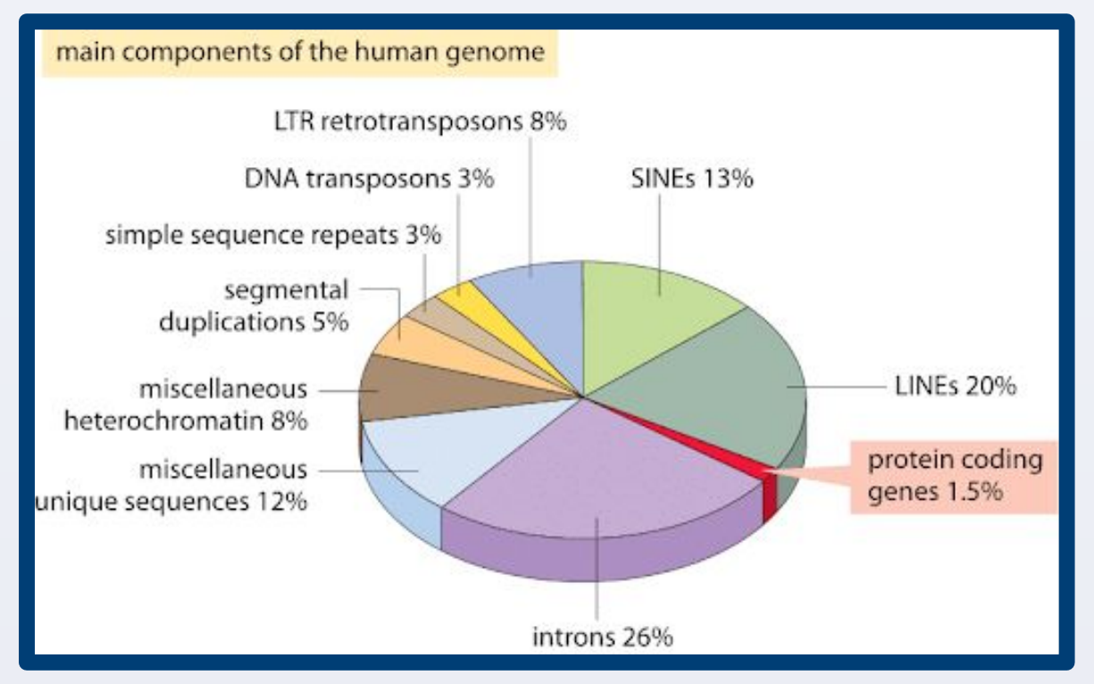
Classifying Coding and Noncoding Regions of the Genome Using Only Sequence Data: A Study using Deep and Interpretable Neural Networks

Shruti Verma, Xuebing Wu
Columbia University Irving Medical Center, The Wu Lab, New York City, New York

Columbia University Medical Center

INTRODUCTION

Only about 2 percent of human DNA is made up of protein-coding genes; the other 98 percent, as has been verified by a multitude of experiments, is noncoding.

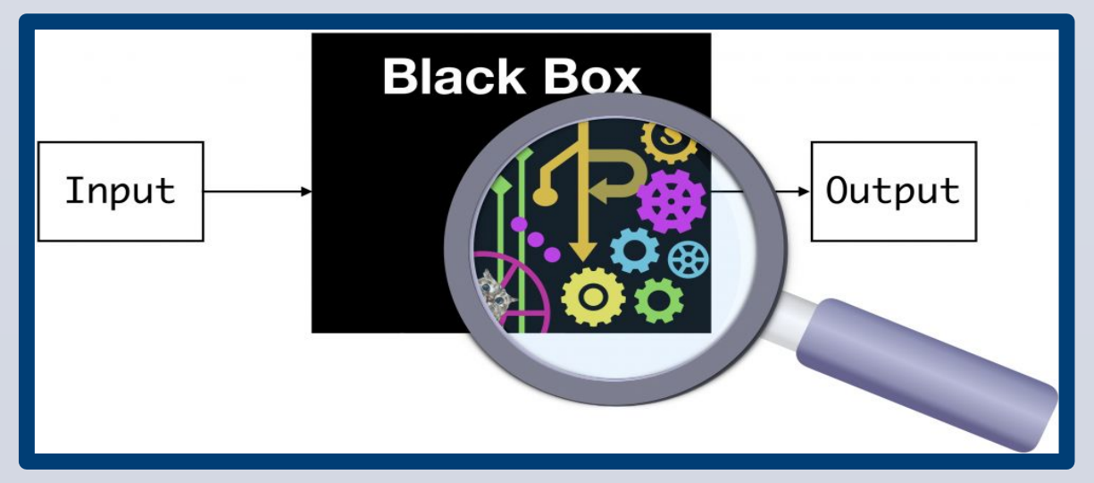


Scientists once thought this noncoding DNA was "junk," with no functional purpose, but groundbreaking research in the early 2010s suggested that this might not be the case. Some segments of noncoding DNA, as it turns out, prove important in regulating gene activity. Knowing this, researchers have been working to understand the location and role of these genetic components.

One method that has gained traction recently is the application of deep and interpretable neural networks to the task of classifying the different portions of the genome. The most successful models that have been developed thus far require various features concerning a given sequence of DNA, such as open reading frame information, codon bias, and more. The Wu Lab set out to investigate whether the coding and noncoding sequences of the genome could be computationally classified using only naive sequence information, in order to better model and thus glean information about biological reality.

ABSTRACT

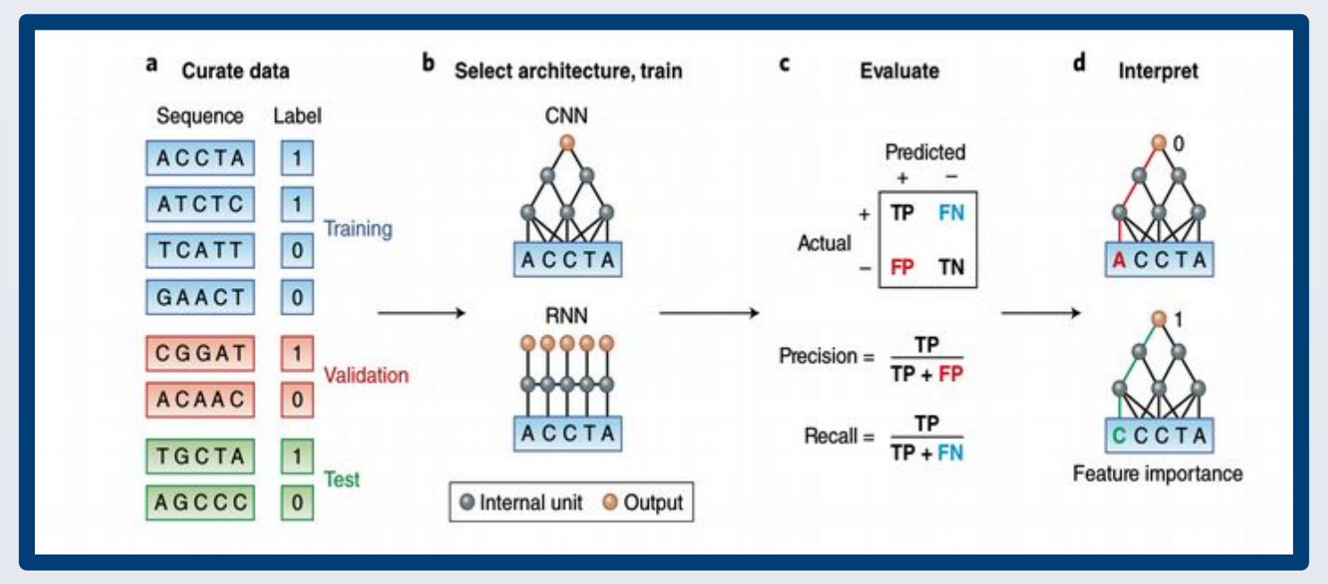
The goal of this project is to learn whether neural network models can be used to accurately identify the coding and noncoding sections of a genome, given only the sequence of nucleotide bases that appear in the segment in question. For this purpose, various model architectures, preprocessing techniques, and evaluation methods were experimented with. Importantly, because this goal stems from a desire to model/understand biological realities as closely as possible, it was important that these networks be somehow interpretable. In particular, as opposed to black boxes, we sought deep neural networks that could indicate what in a given sequence had instructed their decision. Though the project is still in progress, results thus far have been favorable and future steps well established.



MATERIALS USED

- ❖ Server equipped with GPUs and thus able to process large amounts of data and run deep neural networks
- ❖ Human Genomic data (NCBI RefSeq track) for 22 chromosomes from UCSC Genome Browser
  - – Exon and Intron sequences for each chromosome
  - – Exon, Intron, and UTR sequences for each chromosome
  - – Gene annotation data for each chromosome
- ❖ Spyder (open source, cross platform IDE for Python)

METHODS



DATA PROCESSING:

1. Download aforementioned sets of sequences and gene annotation information for each chromosome from the UCSC Genome Browser
2. To construct your training and testing data set, write a preprocessing script in Python to perform the following tasks on each sequence for each chromosome
   - ○ Utilize either the set of exon and intron sequences for each chromosome or the set of exon, intron, and UTR sequences based on whether the user specifies binary/categorical classification
   - ○ Cut sequence to user-specified length (i.e. 100, 300, etc.) so it can be fed through neural network successfully
   - ○ Remove all duplicate sequences based on gene annotation table previously retrieved
   - ○ Construct balanced and imbalance dataset

EXPERIMENTING WITH NEURAL NETWORKS:

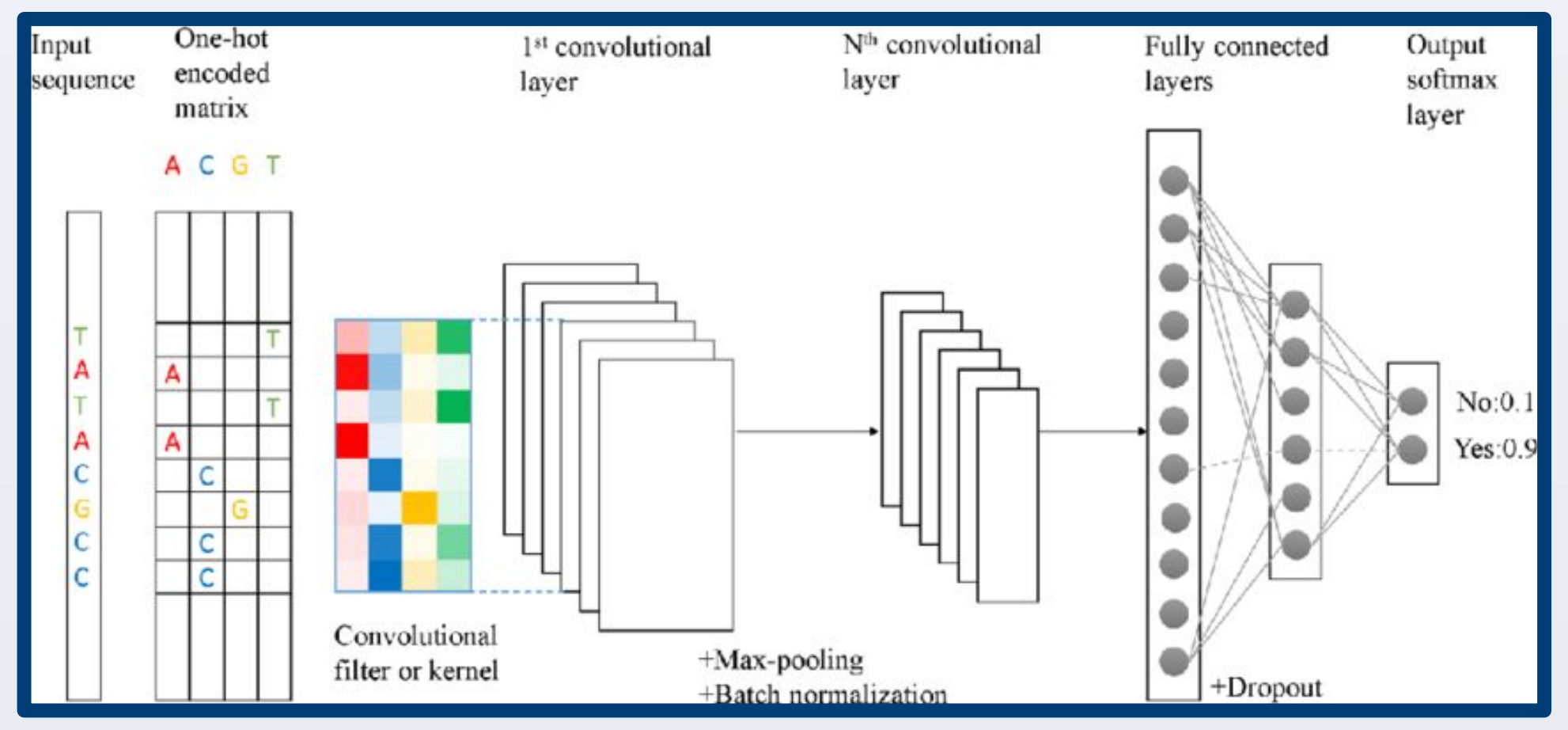
1. Using Tensorflow and Keras, construct neural networks of varying architectures, with a particular focus on Recurrent Neural Networks and Convolutional Neural Networks
2. Train each constructed neural network on a randomly selected portion of the data processed beforehand. Test on the remaining portion of the data.
3. Note the accuracy and tune network hyperparameters until no more improvement seems possible.

UCSC Genome Browser

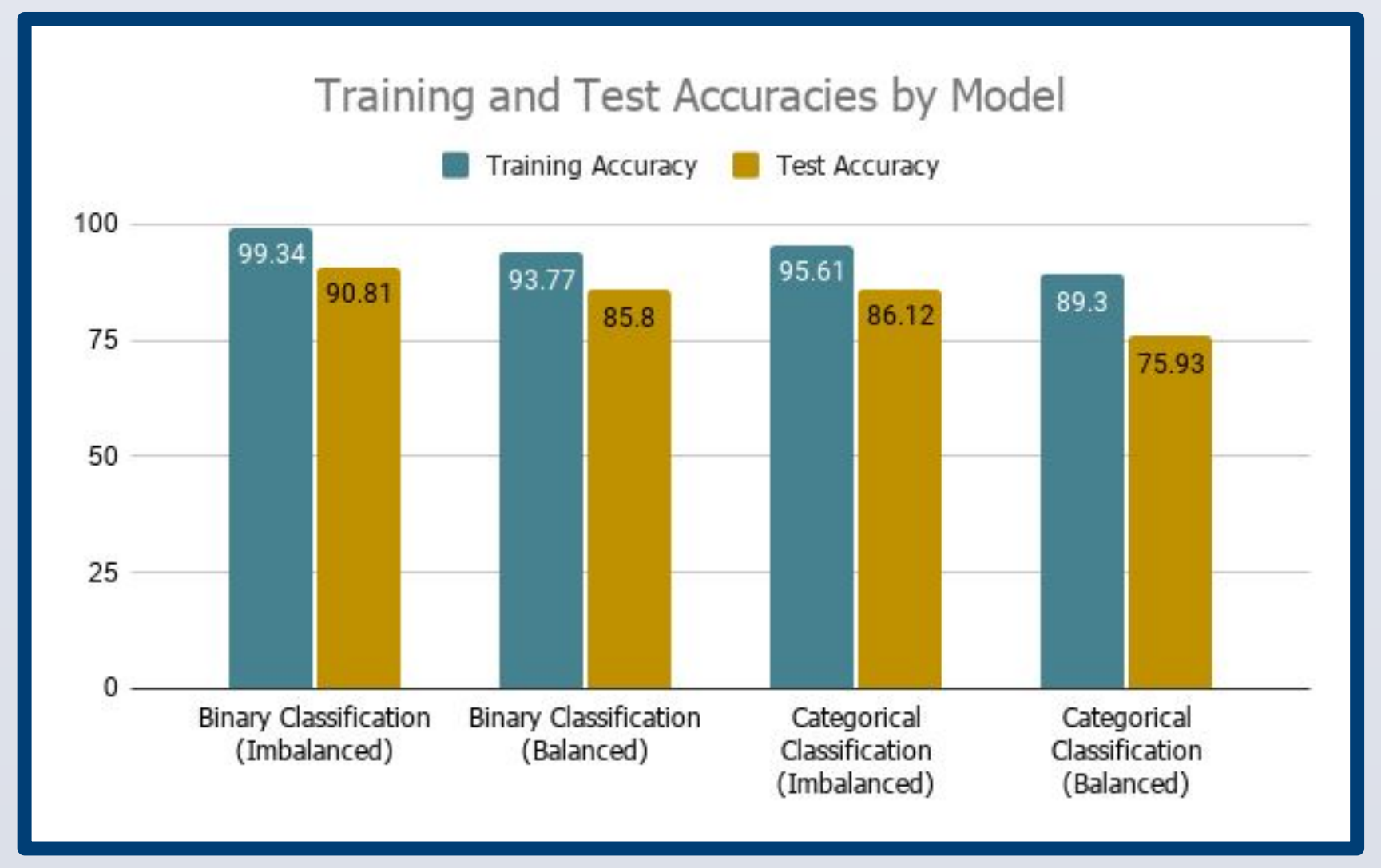
Primary Source of Data

RESULTS

Ultimately, after constructing, experimenting with, and tuning many different architectures, the one that was found to work the best was a Two Dimensional Deep Convolutional Neural Network. Below is a pictorial depiction of the final pipeline used.

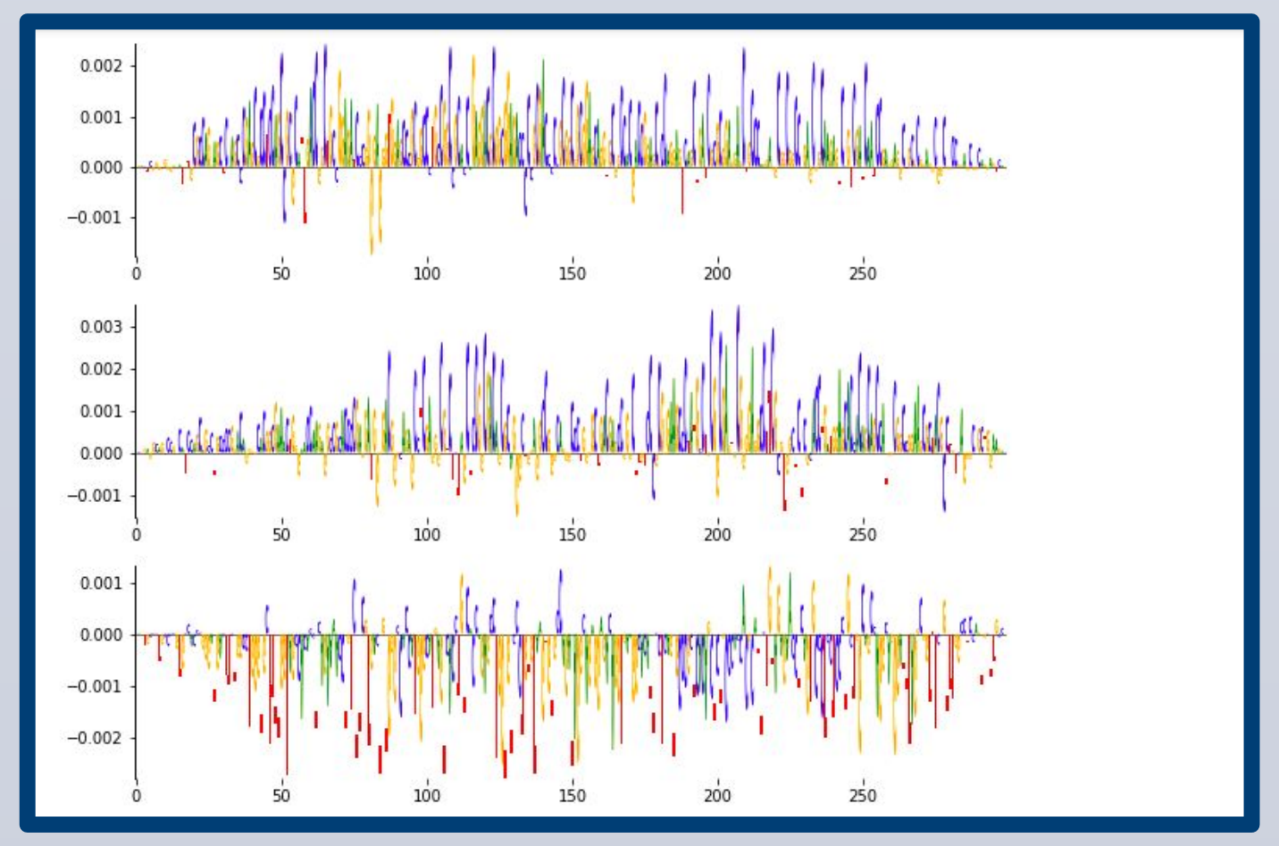


Having chosen and tuned this architecture, we then ran tests on its accuracy in four tasks: binary classification (imbalanced), binary classification (balanced), categorical classification (imbalanced), and categorical classification (balanced). The best scores retrieved are noted in the below bar graph.



Evidently, the imbalance binary classification task was performed most accurately, and the balanced categorical classification task was performed least accurately. Further, there is a steep drop off between each task's training and test accuracy.

Below is an example interpretable result extracted from the binary classification model. Letters facing upwards suggested to the model that the sequence was coding; those facing downwards suggested noncoding. The size of the letters indicates their relative importance in the final decision made.



CONCLUSIONS

The project has not yet reached its final conclusion, but results so far indicate that though it is possible to distinguish between coding and noncoding regions of the gene using neural networks trained only on sequence information, it is rather difficult to do so with very high accuracy and without overfitting to a given training set. As of right now, the accuracies we are getting on the balanced datasets seem close to but not better than the state of the art systems that use multiple features in making their predictions. We hope to continue experimenting with some different and more advanced architectures to see if these are able to remedy the aforementioned problem. We also plan to soon begin analyzing the interpretable results retrieved from the highest performing models to see if, from this, we might learn of previously unknown motifs.

We are also now considering a new avenue of investigation. In particular, we are now interested in learning whether the tools that currently exist on the market for the sake of predicting coding/non-coding sections of the gene can be improved by the results of our models being sent in as an additional feature vector to them. If this is in fact the case, then this implies that though our models may not be able to extract enough information/learn enough patterns from sequence data alone to make accurate predictions independently, they still manage to encode important information regarding the coding potential of a segment of the genome. We are excited to see where this new pursuit leads us.

ACKNOWLEDGEMENTS

I'd like to thank Dr. Xuebing Wu for serving as my primary mentor on this project, as well as Zanis Fang for serving as my point of contact in regards to programming questions. I'd also like to thank everyone in the Wu Lab for welcoming me into their group.

CITATIONS

Chollet, François. *Deep Learning with Python*. Manning Publications Co., 2018.

Eraslan, G., Avsec, Ž., Gagneur, J. *et al.* Deep learning: new computational modelling techniques for genomics. *Nat Rev Genet* 20, 389–403 (2019). https://doi.org/10.1038/s41576-019-0122-6

Jaganathan, Kishore, et al. "Predicting Splicing from Primary Sequence with Deep Learning." *Cell*, vol. 176, no. 3, 2019, doi:10.1016/j.cell.2018.12.015.