Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 26 October 2020
Manuscript Category

OXFORD

Machine Learning

Predicting COVID-19 mRNA Sequence Degradation with a Transformer

[redacted][1,*], [redacted][2] and Naomi Lee[1]

[1]Department of Computer Science, Brown University, Providence, 02912, USA and
[2]Department of Biostatistics, Brown University, Providence, 02912, USA.

*To whom correspondence should be addressed.

Associate Editor: Ritambhara Singh

Received on 30 October 2020; revised on 4 December 2020; accepted on 13 December 2020

Abstract

Motivation: The fight against COVID-19 requires a vaccine that can be effectively transported and widely distributed. mRNA vaccines seem to be the best candidates for immunity against the illness. However, due to the sensitive and spontaneously degrading nature of mRNA, which can render an entire mRNA vaccine useless, it is important to know which part of a given RNA sequence is most vulnerable to cutting and which is safe from degradation. A Transformer lends itself to this learning task because it takes into account the order of an input, i.e. DNA sequences, and has the known performance advantages of deep learning.

Results: A Transformer fit to the DNA sequence data–nucleotide, structure and loop type at each location–predicts the multivariate output of five different degradation measures at sixty-eight different locations along the sequence. This model outperforms the support vector machines used as baseline.

Availability: [redacted]

Contact: [redacted]

1 Introduction

Coronavirus disease (COVID-19) is a highly infectious illness that, since emerging in late 2019, has infected over forty-four million and killed over one million worldwide (Oct. 29 12:30PM GMT). Developing a vaccine is tantamount to controlling the pandemic. The most promising vaccines in development are mRNA vaccines, which deliver synthetic disease mRNA strands to immune cells to prompt the creation of antigens. Unfortunately, mRNA vaccines are extremely fragile. The front-running COVID-19 vaccine would require refrigeration at extremely low temperatures to survive distribution, a costly proposition that will impede access for third-world countries. There is an urgent need to identify locations along COVID-19 mRNA sequences that are prone to spontaneous degradation so that the vaccine can be stabilized.

Deep learning models ingest a large amount of data, and enjoy remarkable performance due to their ability to approximate nonlinear relationships, interaction effects and latent features. Among this class of models, those that learn sequential inputs are particularly well-suited to learning spatial outcomes along vector. We will train a Transformer–a deep learning model popular in natural language processing that learns sequential data without limiting communication to adjacent time-points–on COVID-19 mRNA sequences (approximately 100-bp) to predict different types of degradation.

1.1 Related Work

Transformers are essentially an upgrade over recurrent neural networks (RNNs) [7]. They are appropriate for sequential data, but differ from RNNs in that they don't process input vectors in order. Instead, Transformers use an attention mechanism that allows the model to look at any state prior to its current state. Different states throughout the model draw on the most relevant points in the sequence, whereas in an RNN, later states often do not benefit from earlier states due to the vanishing gradient problem. Transformers are appropriate for genomic data because the context window extends to thousands of base-pairs (or words, in natural language processing). A Transformer-based model has been developed for labeling DNA sequences [2].

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

2 Data

The model uses the COVID-19 Stanford OpenVaccine dataset. Multiple ground truth values are provided in the training data.

2.1 Sequence data

Each input in the dataset is comprised of the following information:

- **sequence**: a 107-length sequence of nucleotides represented by base abbreviation
- **structure**: a 107-length vector giving the predicted molecular structure, showing which bases are paired to each other and which are unpaired
- **predicted_loop_type**: a 107-length vector giving the structural context of each nucleotide, assigned by bpRNA

2.2 Labels

The following columns are provided as the targets for prediction:

- **reactivity**: a 68-length vector giving base reactivity scores for the first 68 base-pairs in the input sequence, used to determine secondary structure
- **deg_pH10**: a 68-length vector giving the likelihood of degradation for the first 68 base-pairs in the input sequence after incubating without magnesium at pH 10
- **deg_Mg_pH10**: a 68-length vector giving the likelihood of degradation for the first 68 base-pairs in the input sequence after incubating with magnesium at pH 10
- **deg_50C**: a 68-length vector giving the likelihood of degradation for the first 68 base-pairs in the input sequence after incubating without magnesium at 50 degrees Celsius
- **def_Mg_50C**: a 68-length vector giving the likelihood of degradation for the first 68 base-pairs in the input sequence after incubating with magnesium at 50 degrees Celsius

Figure 1 shows the marginal distribution of each target for prediction. The targets have no upper or lower boundaries. We experiment with two settings:

1.leaving the values as they are, and
2.standardizing them around a mean of 0.5 and boundaries of 0 and 1.

Scaling to have a 0.5 mean and range between 0 and 1 suits intuition about probability values.

2.3 Augmentative data

The following features are provided for potential augmentation:

- **signal_to_noise**: a singular value denoting the S/N ratio in the sequence, and
- **error columns**: standard error for each measurement of label columns listed above.

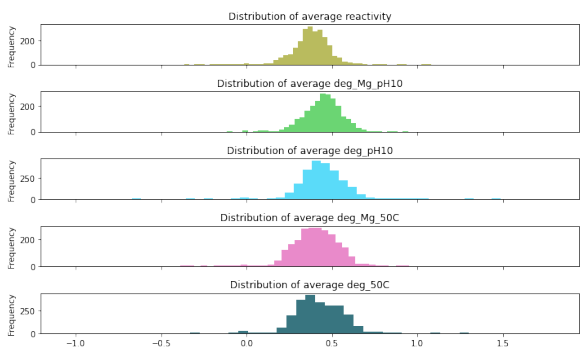

Fig. 1. The distributions of the average for each prediction target are plotted.

3 Methods

3.1 Transformer Model

3.1.1 Transformer

As mentioned previously, the Transformer is a sequence transduction model that implements an encoder-decoder self-attention mechanism [7]. In this setup, the encoder is trained to learn a latent representation of the sequence, whereas the decoder is used to generate tokens at inference time using the outputs of the previous decoder's timestep as well as the encoder's representation.

We will utilize the encoder layer of the Transformer, which produces the latent representations of the mRNA sequence. The output of this can then be mapped to a predictive score.

3.1.2 Hyperparameter search

We perform a grid search on the following hyperparameters for our model:

- learning rate,
- feedforward Dimension,
- sequence embedding dimension,
- number of encoder layers
- and dropout rate.

3.1.3 Architecture

We implement multimodal inputs with multitask outputs in order to learn stronger repesentations and augment performance on each task. Our multimodal inputs, as stated before, are comprised of concatenations of trainable embeddings of sequence, structured, and predicted loop type. We predict, in parallel, each of the five scores of the dataset.

Our model uses a Transformer encoder in order to create strong latent representations of the mRNA sequences and their context. The output of the encoder layer is fed into a dense layer and a subsequent LeakyReLU activation which predicts the degradation scores.

Overall, the following are the steps of our baseline model:

1.Tokenize **sequence**, **structure**, and **predicted_loop_type**
2.Use a trainable embedding layer to embed the tokenized values of each of **sequence**, **structure**, and **predicted_loop_type** into 3 128-dimensional vectors: **embed(sequence)**, **embed(structure)**, and **embed(predicted_loop_type)**.
3.Concatenate **embed(sequence)**, **embed(structure)**, and **embed(predicted_loop_type)** into a vector **x**
4.Encode **x** into a latent representation **h** using a Transformer encoder
5.Input **h** into a dense layer with a LeakyReLU output to determine prediction scores

Gradients are updated according to mean-squared error (MSE) loss.

3.2 Baseline Model

The baseline method is a support vector machine. The targets–$5 \times 68 = 340$ in all, five output measurements for each of the sixty-eight base pairs captured for each observation–are treated as continuous. A support vector machine with a radial kernel is trained on 70% of the 2400 observations for each target. The remaining 30% of observations are used as a held-out validation set.

4 Results

Our main metric is the measured MSE loss on our test sets. Furthermore, we provide ROCAUC data for the Transformer model, which shows the strength of the model as a binary classifier to determine whether each nucleotide experiences degradation higher or lower than the median degradation.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

4.1 Baseline model

Table 1 shows the mean squared error (MSE) on training and validation sets under a variety of settings. Every MSE is averaged over all targets.

As shown, the MSE for the baseline model hovers between 0.77-0.88.

MSE	Targets	Covariates	Evaluation Set
0.77	Raw	All	Train
0.86	Raw	All	Held-Out
0.77	Raw	Sequence-Only	Train
0.87	Raw	Sequence-Only	Held-Out
0.85	Standardized	All	Train
0.87	Standardized	All	Held-Out
0.86	Standardized	Sequence-Only	Train
0.88	Standardized	Sequence-Only	Held-Out

Table 1. The baseline method is fit using raw or standardized target values, and all covariates (sequence, structure, loop typed) or only sequence data. The lowest MSE achieved on the held-out validation set is for raw target values and all covariates.

4.2 Transformer

We hypothesized that training our model to predict degradation along COVID-19 mRNA sequences will deliver results competitive against the current standard in machine learning with reduced training time and improved loss compared to the SVM.

In our preliminary experiments, we measured the effect of the contextual data (sequence and predicted loop type) on our model's performance. The addition of this data resulted in a decrease of 0.02 MSE test loss on our baseline Transformer model.

In our experiments using the sequence of nucleotides alongside the sequence and loop type data, we achieved an ROCAUC score of 0.81 with 0.73 MSE loss using our optimized Transformer model. For context, across hyperparameter experiments our ROCAUC stayed between a range of 0.6 to 0.81.

4.3 Hyperparameter search

Moreover, our hyperparameter search gave us strong insights into the performance limits of this architecture on the given data as prepared.

Our search revealed the following hyperparameters as optimal:

- Learning rate = 0.0005
- Model dimension = 256
- Embedding dimension = 64
- Dropout = 0.01
- Num encoder layers = 2

Notably, changing dropout and learning rate from these optimal values caused high variation in results, up to a difference of 0.07 test loss.

When adding more parameters to the model beyond a model dimension of 256 with 2 encoder layers, it appears that the model becomes overparameterized and performance plateaus or, in some cases, becomes slightly worse.

5 Discussion

The open question of degradation rates along COVID-19 mRNA sequences is well-suited to deep learning methods that account for ordered inputs, which are powered to detect nonlinear relationships and interaction effects. Joint estimation of five measures of degradation at once leverages the outputs' covariance structure.

The five target vectors have unbounded values giving scores for degradation under different conditions. An alternative formulation of the problem is to transform the output to $\{0, 1\}$ using 0.5 cut-off which, although a loss of information, would allow us to experiment with different objective functions.

This work supports efforts to stabilize a refrigerator-stable COVID-19 vaccine. To that end, it will be beneficial to summarize the five estimates at each location to identify the locations that are most broadly fragile.

Our model, which predicts RNA degradation for relatively short molecules (between 50-150 nucleotides), may be used for significantly longer RNA data or full-length mRNAs (e.g. 3000-400 nucleotides), which parallels those currently used in the COVID-19 vaccines.

There are several comparable deep learning architectures with could be used to deal with longer genomic sequences. The most obvious are offshoots of the traditional Transformer architecture. These are motivated by computational limitations of the original Transformer, e.g. a sequence of length $100,000$ means that $100,000 \times 100,000$ pairwise relationships will be considered, and hidden layer outputs will take up a huge amount of memory. More efficient variations on the Transformer include the Reformer, a deep learning architecture for sequential inputs that has an advantage over the state-of-the-art because it is able to glean more information from longer inputs with greater efficiency [5].For example, the Compressive Transformer remembers past activations by "compressing" them into coarse memories [6]. An older Transformer, the TransformerXL, remembers past activations until they are sufficiently old, at which point they are discarded [3]. Another Transformer, Big Bird, ingests long sequences with the introduction of an attention mechanism that is linear in the number of tokens, rather than quadratic [8]. Finally, a traditional deep learning architecture that is older and less optimized, but proven, is a recurrent neural network [1, 4].

One other possible improvement is to use graph connections and a message-passing neural network to model the genomic structure, instead of passing in the structure data as a sequence to the model. This will allow richer interactions between connected cells (nodes in the graph) to be modeled.

6 Contribution

DA suggested the Transformer architecture and fit the deep learning models using Python. DU fit the baseline support vector machine models using R. NL found the data, conceptualized the research question and led write-up and presentations.

References

[1]Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[2]Jim Clauwaert and Willem Waegeman. Novel transformer networks for improved sequence labeling in genomics. *bioRxiv*, 2020.

[3]Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860, 2019.

[4]Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[5]Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer, 2020.

[6]Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019.

[7]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

[8]Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences, 2020.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture