MISE RESEARCH PROGRAM

PREDICTING FUTURE MUTATIONS OF COVID-19

KEITH KOJO TORPEY

RESEARCH MENTOR: MATTHEW SCICLUNA

Abstract

The pandemic brought about by Covid-19 has greatly disrupted the world. The virus has different mutations that can be seen in the population, but there are also many mutations that exist within a person, not seen in the population. These intra-host variations of the virus can be quantified using variant allele frequency (VAF) data. With deep learning processes, we tested the hypothesis that VAF data can be used to predict future mutations of the virus.

We found the difference in allele frequencies between the first and second waves of Covid for each base position. We set thresholds for the difference in frequencies that could be considered as a mutation. We then binarized our targets using these thresholds and determined whether a base position mutates or not. We then used PyTorch and Google Colab to train a deep learning system that took the VAF data as an input and matched it to the targets in an attempt to predict whether a base position mutates or not using VAFs.

The system was tested against a portion of the dataset that was held out, and the results were analyzed. The results obtained showed that the model could not efficiently predict the positive cases where a base position mutates. Across different thresholds the results for the accuracy, precision and recall varied, ranging from 97.5%, 42.9% and 52.8% to 51.6%, 1.9% and 2.5% respectively.

The study did not put forward conclusive evidence to the research hypothesis that VAF data could predict future mutations. Further research could be done to explore the hypothesis with changes made to the VAFs to improve the data.

1 Introduction

The Covid-19 pandemic is one that has ravaged the world with its devastating effect on all aspects of life, from health to the economy. There have been over 190 million reported cases and over 4 million deaths as a result [1]. With how massive its impact is, there is a growing need for innovative and scientific research to further understand the virus. And in this effort, thanks to the continuous effort of many researchers, we can use viral sequence data this endeavor to stop Covid.

New variants are bound to appear due to mutations of the virus's genes, and these variants can come with different effects, and may cause the virus to become more deadly and/or more contagious. Some new variants can provide increased vaccine resistance and they would be selected for in the general population as they are more likely to survive. For example, early data shows that the Oxford-AstraZeneca Covid-19 vaccine provided "minimal" protection from the B.1.351 coronavirus variant [2]. If a mutation that provides high vaccine resistance emerges and becomes prevalent in populations, then a new vaccine would be needed to combat it. This raises the need for genomic surveillance in order to predict future mutations [3].

Intra-host variation can be an essential tool in predicting mutations of the virus, as mutations which exist within an individual may eventually be seen in the population. Understanding the evolutionary dynamics of Covid within a person can help to give an insight into the more general global mutations. We proposed to use deep learning methods along with intra-individual mutation data to predict which strains are likely to appear in the general population.

1.1 Literature review

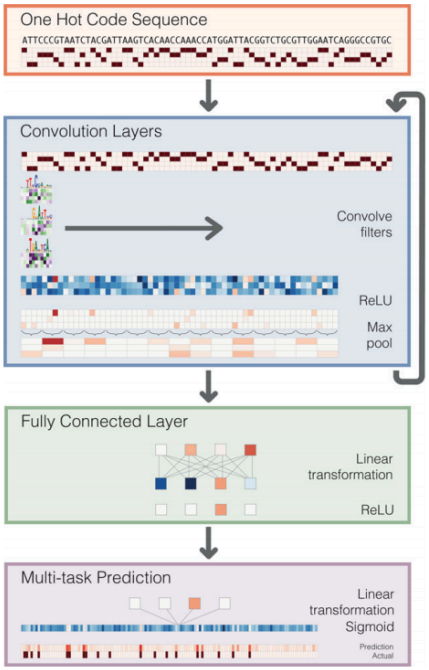
1.1.1 Variant allele frequency

The numerous strains which can appear would be as a result of mutations and changes in the RNA sequence. The Variant Allele Frequency (VAF) is

defined as the number of variant reads(the most frequent alleles which are not the reference) divided by the number of total reads and this can be represented in a percentage form [4]. Our reference genome would be from the first strain sequenced in Wuhan and we will use SARS-CoV-2 RNA sequence data downloaded from the NCBI database, as well as the list of worldwide mutations and variant frequencies obtained from Gisaid, an organization promoting the rapid sharing of COVID-19 and other data. The data is collected from patients and would measure the degree of variation seen within an individual. The more reads of sequences that deviate from the reference genome, the greater the VAF would be. A total of 12404 Illumina SARS-CoV-2 virus sequencing reads were downloaded from the NCBI database to be used.

1.1.2 Machine learning

An artificial neural network [5] is a learning algorithm which is used to find relationships within a data set. They are designed to imitate the processes of the brain in analyzing and in pattern recognition. There are various kinds of neural networks such as convolutional neural networks, feed forward networks and recurrent neural networks. We employed a simple feed forward network consisting of 3 layers, an input layer, the hidden layer and the output layer. Neural networks are widely used in a variety of applications, including in the genomics context, with an example being Basset [6], which uses a Convolutional Neural Network (CNN) to learn the functional activity of DNA using genomic sequence data.



Keith Torpey-MRP 2021

This is the architecture of Basset, with components like the Convolution layers and a fully connected layer. It is a deep convolutional neural network (CNN) for DNA sequence analysis. Basset predicts the cell-specific functional activity (with an example being DNase I hypersensitivity) of sequences.[6]

1.1.3 Learning Process

We trained a neural network with the VAF data to predict the difference in mutations between the first and second waves of Covid. The neural network was trained to recognize patterns in using a training set which was a larger fraction of our data and tested against the rest. The project was done using Google Colab and implemented in PyTorch [7].

Once finished training the program would attempt to follow the pattern and predict whether certain gene positions are more likely to mutate in the future. We used a validation set to assess the system as it was training, and to determine whether to use other regularization methods like early stopping. The future performance was then evaluated using a held-out test set.

1.2 Purpose

Our hypothesis is that the VAF data can help us predict mutations of Covid that will appear in the future. To the best of our knowledge, VAF data has not been used in any deep learning network.

Our aim was first be to identify possible batch effects brought about from sequencing centers, and to get rid of them. Our next aim was to train the system to identify strains that are likely to break out into the population using the data.

This is essential because once we can predict future variants which may have vaccine resistance, then the existing vaccines could be modified or new vaccines could be developed ahead of time which would be able the combat these new variants, and hence the next wave would quickly be curbed and prevent a new surge of cases and a more deadly form of the virus.

2 Methodology

Our goal was to build a neural network which can be used to determine which base positions in the RNA sequence of Covid are likely to mutate. We hypothesized that using the intra host VAF data can be used to predict which positions mutated globally between the first and second wave of Covid.

2.1 Neural network

We used PyTorch and Google Colab to create the neural network to be used in this project. It was a Feed Forward Network which takes the VAF data as an input and compares it with specified targets. We created a neural network which has 2 linear layers and varied various conditions.

We decided to binarize our targets, since we are trying to observe whether a base position would mutate or not. Due to this we used the Binary Cross Entropy as our loss function. We initialized this using

```
critereon = nn.BCELoss()
```

We used the Adam optimizer, here we provided the learning rate for the model, and it would be varied to test different models. We also applied a weight decay in the optimizer to prevent overfitting of the model.

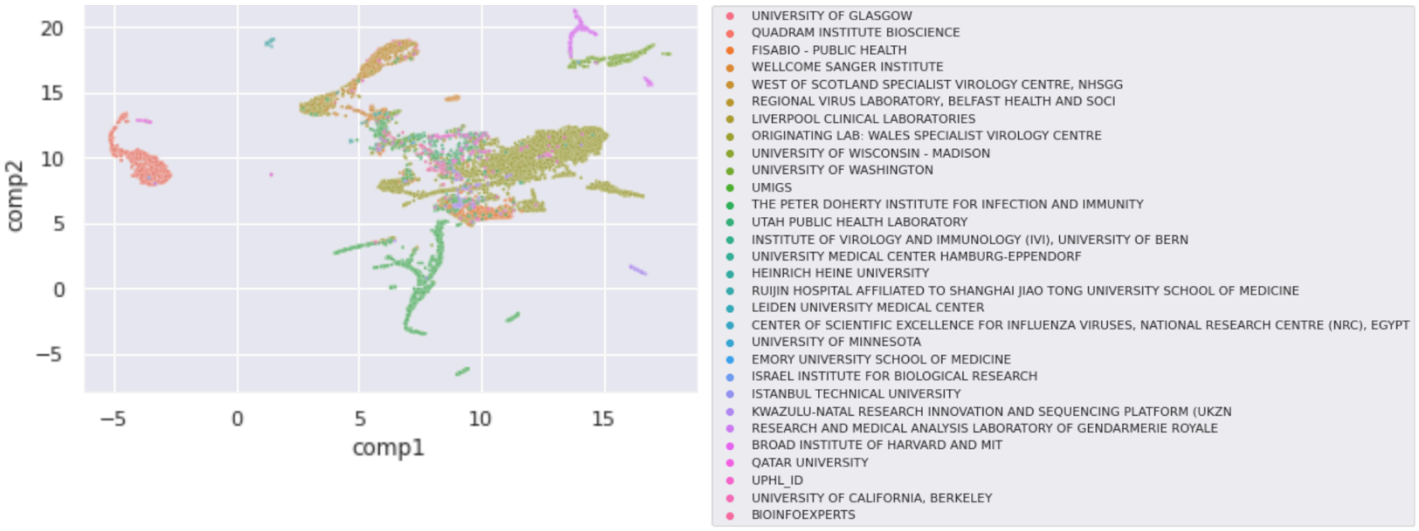
```
optimizer = optim.Adam(model.parameters(), lr= , weight_decay=0.001)
```

The aim of this neural network was to take in VAF data from the individuals sampled and use this the targets, which are mutations between the first and second waves of Covid, to learn whether a position would mutate or not based on the VAF data. We split the VAFs and targets into a training set for the model, a validation set and a test set to aid in evaluation.

2.2 Data Processing

Vaf Input

As mentioned, our input was the VAF data collected, so first in processing the data we tried to remove any batch effects that could potentially affect the results and effectiveness of the model. For the VAFs lower than 5%, there appeared to be some batch effects based on the sequencing center, and this could be due to sequencing errors. We performed a PCA and then a UMAP [9] to demonstrate this. We colored the points in the UMAP according to the sequencing center. The UMAP plot of the VAF data with values below 5% was:



From this we noticed that most clustered points all have the same color, and this was most likely due to sequencing errors based on the center. To get rid of these batch effects, we considered setting all the VAFs below 5% to 0, so that they would not affect the results. But with implementation of this, the majority of the VAF values (over 99%), were 0 so the signal for training the model was weak. As a result, we left the VAFs below 5%.

We removed the VAFs which are 0 as well as their corresponding targets as they are not needed for the model.

Targets

To create our targets, we first found the nucleotide frequencies for the first and second waves of covid at each position. The arrays of frequencies were created using the code below.

```
# Empty arrays to be filled with A,C,T,G frq
first_wave = np.zeros((df1.shape[0], 4)) # nb of positions x 4 nucleotides
second_wave = np.zeros((df1.shape[0], 4))
alleles = ['A', 'C', 'T', 'G']

# Fill arrays
for i in range(df1.shape[0]):
  # --- First wave ---
  obs_alleles = []
  dict_frq = dict()
  if type(df1.iloc[i]['A1_frq']) == str:
    obs_alleles.append(df1.iloc[i]['A1_frq'][0])
    dict_frq[df1.iloc[i]['A1_frq'][0]] = np.float32(df1.iloc[i]['A1_frq'][2:])
  if type(df1.iloc[i]['A2_frq']) == str:
    obs_alleles.append(df1.iloc[i]['A2_frq'][0])
    dict_frq[df1.iloc[i]['A2_frq'][0]] = np.float32(df1.iloc[i]['A2_frq'][2:])
  if type(df1.iloc[i]['A3_frq']) == str:
    obs_alleles.append(df1.iloc[i]['A3_frq'][0])
    dict_frq[df1.iloc[i]['A3_frq'][0]] = np.float32(df1.iloc[i]['A3_frq'][2:])
  if type(df1.iloc[i]['A4_frq']) == str:
    obs_alleles.append(df1.iloc[i]['A4_frq'][0])
    dict_frq[df1.iloc[i]['A4_frq'][0]] = np.float32(df1.iloc[i]['A4_frq'][2:])
```

Doing this for both the first and second waves allowed us to compute the differences in the frequencies observed for each base position.

We computed our targets with the formula

Abs_(Base in {A,C,U,G}(|rate(base, first wave) – rate(base, second wave)|))

With this difference, we created our binary condition by selecting thresholds of what is “sufficient” mutation. We used 4 different thresholds and the first was $2e^{-5}$- where about half of the bases are positive and the other half is negative. The other 3 thresholds were 0.0001, 0.001 and 0.005. These thresholds were selected as they are greater than $2e^{-5}$, so there would be a lower number of mutations, and hence may more accurately depict the number of true mutations which occur.

These targets were what the VAF data was compared to, to learn which values of VAF brought about the mutations.

2.3 Training and manipulations

The neural network was trained multiple times with different seeds to find average results across varying conditions. This was then repeated this with the 3 other target thresholds .

We also applied weighted training for these cases, giving more weight to the positive cases, so the model does not learn to only predict 0.

We repeated all this with a model that contains one linear layer instead of two layers.

3 Results

We trained 2 different neural networks, one with 2 linear layers and 1 with a single layer and tested these both across the 4 different thresholds of what is considered sufficient mutation between the first and second waves. In order to better understand our model performance, we computed the accuracy, precision and recall of each variation of the model to aid in analysis. The precision was the accuracy of our positive predictions and the recall would be how many positives we predicted.

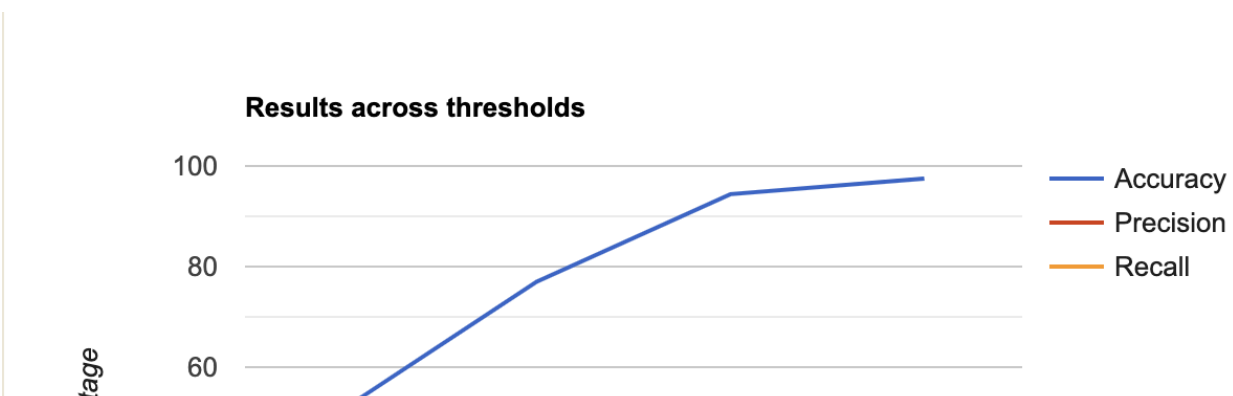
The network trained with 2 layers did not show much learning and underfit, so we opted to focus on the results of the network trained with a single layer.

3.1 Analysis

We trained the system for each threshold 5 times with different seeds and we found the average value across these tests.

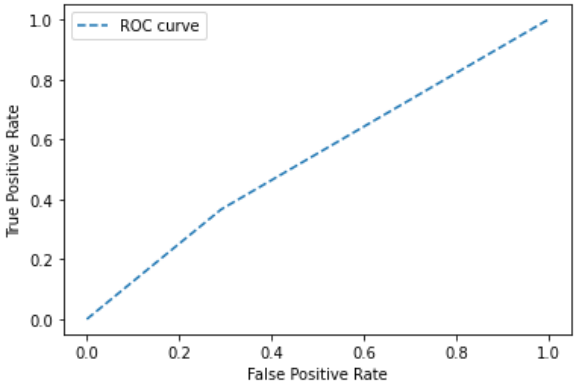
The table and graph below show the average accuracy, precision, and recall across the 4 thresholds for the Neural Network with one layer, with the standard deviation for each metric shown as well.

Evaluation Metrics/(%) (+/- SD)		**Thresholds**			
		$2e^{-5}$	**0.0001**	**0.001**	**0.005**
	Accuracy	51.6 (2)	77 (0.6)	94.4 (0.2)	97.5 (0)
	Precision	49.2 (2)	25.6 (2)	4.6 (3)	1.9 (0.5)
	Recall	52.8 (1)	20.0 (8)	2.5 (1)	3.9(1)



We notice that as the thresholds for the targets increase, the accuracy of the model also increases, but both the precision and recall fall. This could be that the model is mostly learning to predict 0 and cannot accurately identify the positive cases, and this can be due to various reasons.

An example ROC curve for the model was :



The ROC plots and AUC shows that the model is not efficient, as the ROC curve lies close to the diagonal, with an AUC value near 0.5. For the $2e^{-5}$ threshold, the mean AUC of the ROC curve was 0.52. All the other thresholds had a lower value. We also found that the average spearman's correlation coefficient between the true data values and what was predicted by the model to be below 0.1. This demonstrates that the model was not efficient.

The results gotten from the models do not seem to justify or give evidence towards the research hypothesis that VAFs can be used to predict future mutations, but this could also be due to other variables so further trials could be necessary.

Conclusion

We trained a neural network with the aim of testing the hypothesis that VAFs could be used to predict future mutations of Covid. The network could predict with high accuracy under some thresholds but was not efficiently predicting positive cases. The results did not give much support to the hypothesis, but further improvement tests could be made to explore the claim.

Limitations and Future Improvements

The VAF data still contained some batch effects due to the sequencing center and these VAFs below 5% made up a significant number of the non-zero VAFs, so this could have affected the performance of the model significantly.

The model could further be improved in the future by possibly correcting the batch effects using methods like replacing VAFS below 5% with an averaged value instead of 0. The model could also other regularization methods, as well as testing new thresholds, and different weights to the positive and negative cases.

Citations

1. Worldometer, Coronavirus Cases, https://www.worldometers.info/coronavirus/,
2. Johns Hopkins Medicine, New Variants of Coronavirus: What you should know, https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/a-new-strain-of-coronavirus-what-you-should-know,
3. Center for Disease control and Prevention, What is Genomic Surveillance? https://www.cdc.gov/coronavirus/2019-ncov/variants/variant-surveillance.html
4. Wiley Online Library, The prognostic impact of variant allele frequency (VAF) in TP53 mutant patients with MDS: A systematic review and meta-analysis, https://onlinelibrary.wiley.com/doi/10.1111/ejh.13483
5. IBM, What are Neural Networks? https://www.ibm.com/cloud/learn/neural-networks
6. David R. Kelley, Jasper Snoek, and John L. Rinn, Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4937568/pdf/990.pdf
7. ArXiv.org, PyTorch: An Imperative Style, High-Performance Deep Learning Library, https://arxiv.org/abs/1912.01703, 3/12/2019
8. Center for Disease control and Prevention, About Variants of the Virus that Causes COVID-19, https://www.cdc.gov/coronavirus/2019-ncov/variants/variant.html

9. Leland McInnes, John Healy, James Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.” *ArXiv.org*, 18 Sept. 2020, arxiv.org/abs/1802.03426.