DeepVariantFinder: Detecting Novel SARS-CoV-2 Variant Genomes using Autoencoders

Mihir Bafna, Vikranth Keerthipati, and Ruhan Ponnada

1 Introduction

Amidst the coronavirus pandemic, the rise of new variants—many being more transmissible and contagious (i. e. Delta, Omicron)—has prompted much discussion regarding the early detection and classification of these variants. Early detection is crucial for preventing rapid transmission but, how do we go about doing this? If we wanted to figure out whether or not a DNA sample of the virus were to be classified as a different strain or variant, a simple pattern match or sequence alignment between the two genomes could be enough to determine the difference. But, how much of a difference is enough to be considered a new strain? Instead, what if we could use deep learning to determine different structural or pattern differences of the viral genomes?

Now that we have established our motivation for using deep learning, we will explain our use of autoencoders. Autoencoders are a specific class of neural networks designed to learn lower dimensional latent embeddings of input data in an unsupervised manner. The autoencoder consists of a series of encoding layers that reduce the dimension of the input followed by decoding layers that try to reconstruct the original input. During training, the model is optimized on the loss of how well it can reconstruct the input from the latent embeddings. This trained model serves two purposes: the encoding layers of the model can be used to map the input genome sequences into lower dimensional embeddings (dimensionality reduction) and the decoding layers can be used as a generative model. One route we could take would be to utilize the decoding layers and generate new valid viral genome sequences, but the latent embeddings are what we are really after. These lower dimensional representations of the viral genomes can bring to light subtle differences between variants and thereby even predict potential viral emergence [3].

Now, we can outline our procedure and methodology: First, we gathered SARS-CoV-2 genomes and pre-processed them to be valid inputs into the autoencoder we developed; Next, we trained the autoencoder on the strains that we have more data on (Alpha and Delta); Then we passed Alpha, Delta, and also Omicron variant genomes through our encoder to obtain latent representations; We visualized these embeddings using PCA and saw that Omicron genomes did indeed form a separate cluster. Finally, we trained a simple LightGBM classifier on the latent embeddings and it was easily able to distinguish between the variants.

2 Methods and Results

In this section, we describe our methods of (2.1) pre-processing the genomic data, (2.2) creating autoencoder architectures, and (2.3) distinguishing novel SARS-CoV-2 variants via PCA and LightGBM. This will be paired with metrics of our model's classification task.

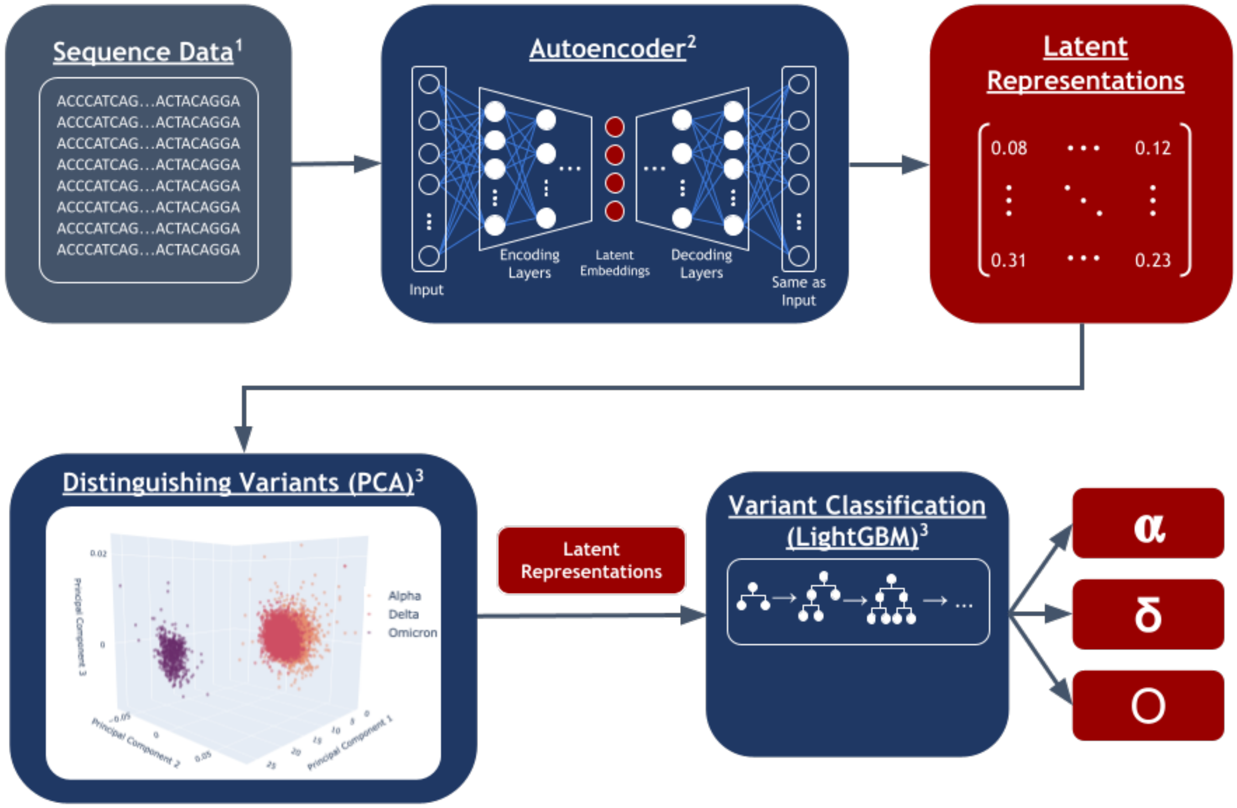

Figure 1: DeepVariantFinder Pipeline. Input sequence data is preprocessed (Seq2Vec) and then passed into an autoencoder in which the encoding layers output latent embeddings of the genomic data. These latent embeddings can be visualized (PCA) and accordingly classified (LightGBM) to understand the emergence of new variants (Alpha, Delta, Omicron, etc). Dark blue colored boxes are denoted as model architectures and red boxes are model outputs.

2.1 Data and Pre-processing

SARS-CoV-2 genomes are 29,000 base pairs long where each base pair is one letter $\in \{A, C, G, T\}$. This means that there are 29000^4 possibilities of DNA sequences for the coronavirus. But, this does not account for the structure and order of the genomes. This is why in the context of genomes, we consider k-mer (sequences of length k). With this, we can encode the frequency of the of each k-mer and represent these frequencies as a vector. This vectorization is crucial as it allows for an input with a size independent of the length of a genome.

The model was trained using a dataset containing complete genome sequences of SARS-COV-2, with the specific variants of B.1.1.7 (Alpha), B.1.617.2 (Delta), and B.1.1.529 (Omicron). This dataset originated from the GISAID initiative [1][4] and was categorized by variant. The training data for the autoencoder consisted solely of alpha and delta variants. For testing classification, we utilized all variants. The dataset consisted a total of 30,934 samples, with 17,373 samples being the alpha variant, 13,109 samples being the delta variant, and 452 samples being the omicron variant.

2.2 Autoencoders and Latent Representations

To obtain our latent representations, we tested various autoencoder architectures. Our first thought was to use a convolutional autoencoder approach with 1 dimensional convolutional layers and k being the filter size which would mimic the k-mers of a genome. In theory, the filter maps outputted would be the most highly activated learned k-mers, which could be significant in determining differences between variants. Convolutional layers would also simplify the amount of trainable parameters necessary rather than a dense layer approach. However, the latter turned out to be more favorable as if the k-mer size was too high, the number of filters would increase as well. We ended up settling on the dense layer architecture, which was much simpler.

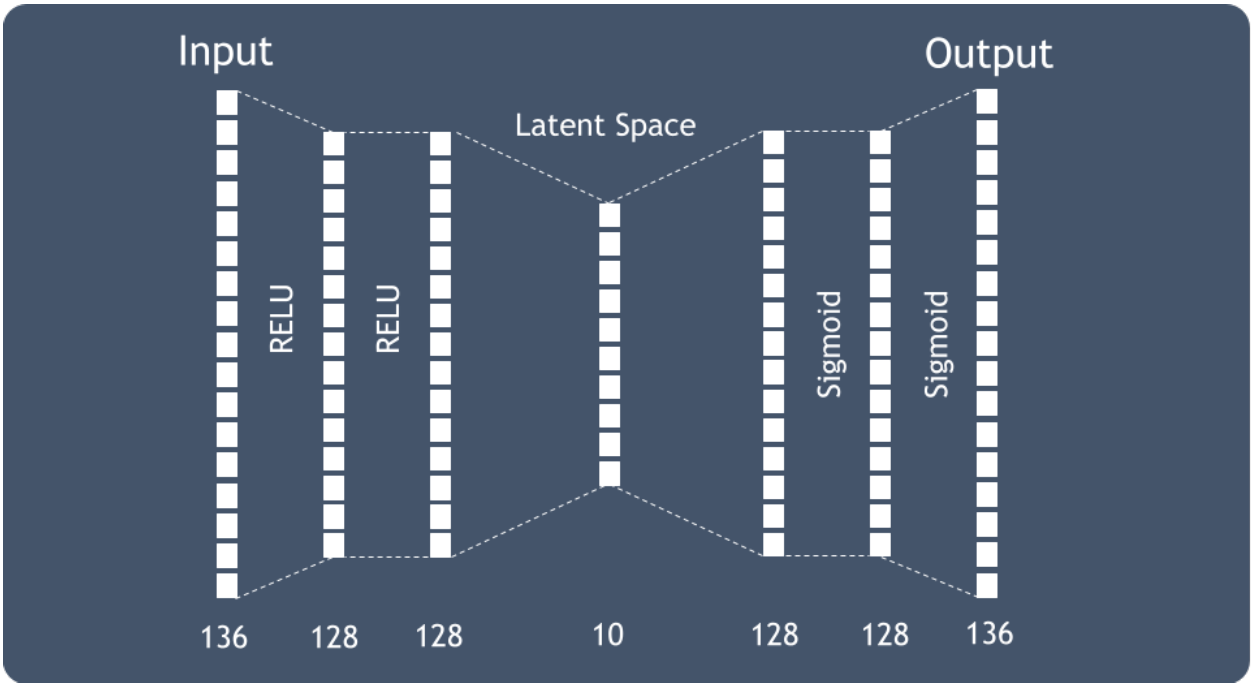

Figure 2: Dense Layer Autoencoder Architecture

To not lose any information regarding the k-mers, we used the Seq2Vec library and mapped all of the genomes to k-mer vectors (explained in 2.1). The encoder consisted of an input layer, two dense layers of size 128, and finally the latent embedding layer, which we decided to be 10 dimensions. Each dense layer was followed by a rectified linear unit (ReLU) activation function. The input of the encoding half would be the vectorized genomes x. The encoding part of the architecture can be written as the following function:

$$E(x) = ReLU(Dense(ReLU(Dense(Input(x)))))$$

The decoding half of the autoencoder is exactly the mirror image of the encoding half, except sigmoid functions in place of the ReLU activation functions. The input to the decoding half of the autoencoder would be the latent embeddings y.

$$D(y) = \sigma(Dense(\sigma(Dense(y))))$$

We trained this autoencoder architecture on 30,482 genomes comprised of only the Alpha and Delta variants as these are much more common than the others. In this way when we pass the Omicron genomes through the encoding layer and obtain the latent embeddings, it simulates how we would do the same for a novel variant in real life (as our model has never seen it). We used Mean Squared Error loss on the model's output compared to the original input, as the goal for the model is to reconstruct the input from the latent embeddings. Once training is complete, we passed all the variants (Alpha, Delta, Omicron) through the encoding layers and obtained the latent representations. These representations were then used to distinguish between the variants.

2.3 Distinguishing Variants and Performance

To distinguish between the variants we utilized another dimensionality reduction step: PCA with 3 principle components (from 10 latent dimensions to 3) in order to visualize on a three dimensional graph. The PCA is shown in Figure 3. Clearly, the Omicron variant forms a distinct cluster, therefore we have shown that new variants can be distinguished and identified simply by encoding their genomes into these latent representations.

To further our analysis, we trained a LightGBM model to learn the differences between the variants and be able to classify them accordingly. LightGBM (Light Gradient Boosting Machine) is an industry standard that takes an ensemble of decision trees and instead of parallel wise training, it takes a more sequential approach with leaf-wise growth. It performs great for most classification tasks, which is why we used it for this latent representation classification problem.

The LightGBM model was trained on 17,373 Alpha latent representations, 13,109 Delta latent representations, and 452 Omicron latent representations. The train to test split was 70–30. All of the genomes used for obtaining the latent representations were from the GISAID[1][4] database. Our model classified 74.09% alpha correct, 89.98% delta correct, and 100.00% omicron correct. With some hyperparameter tuning, this could be improved, however it was enough to explain the fact

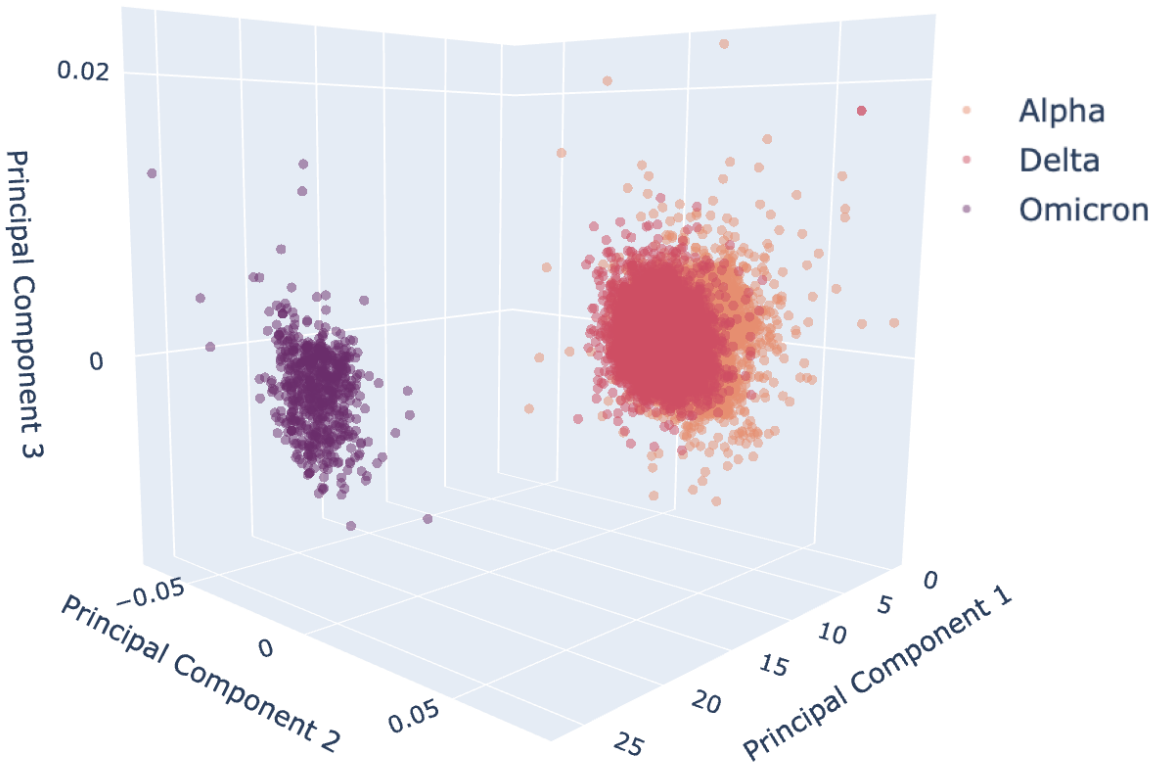

Figure 3: PCA on Latent Embeddings The omicron variant latent embeddings (purple) form its own distinct cluster from the other variants.

that these clusters between the "known variants" (alpha and delta) and the "unknown variant" (omicron) are highly separable. This can now be transferable to any new genome. We simple just pass it through the encoder and if the latent embeddings of that genome appear much further away from any other cluster, we could classify it as a variant.

3 Baseline and Discussion

As we can see from the PCA from Figure 3, the latent embeddings are highly separable from variant latent embeddings. This poses the question that the data might be highly separable without the lower dimensional representation. This could mean the autoencoder's latent representations are not capturing anything particularly meaningful as the variant genomes are separable themselves. To test this, we developed a baseline test. We simply just visualized the PCA of 3 components of the original Seq2Vec vectorized genomes (not the latent representations). This PCA is shown in Figure 4. As we can see, there are indeed clusters of the variants, however it is nothing compared to the

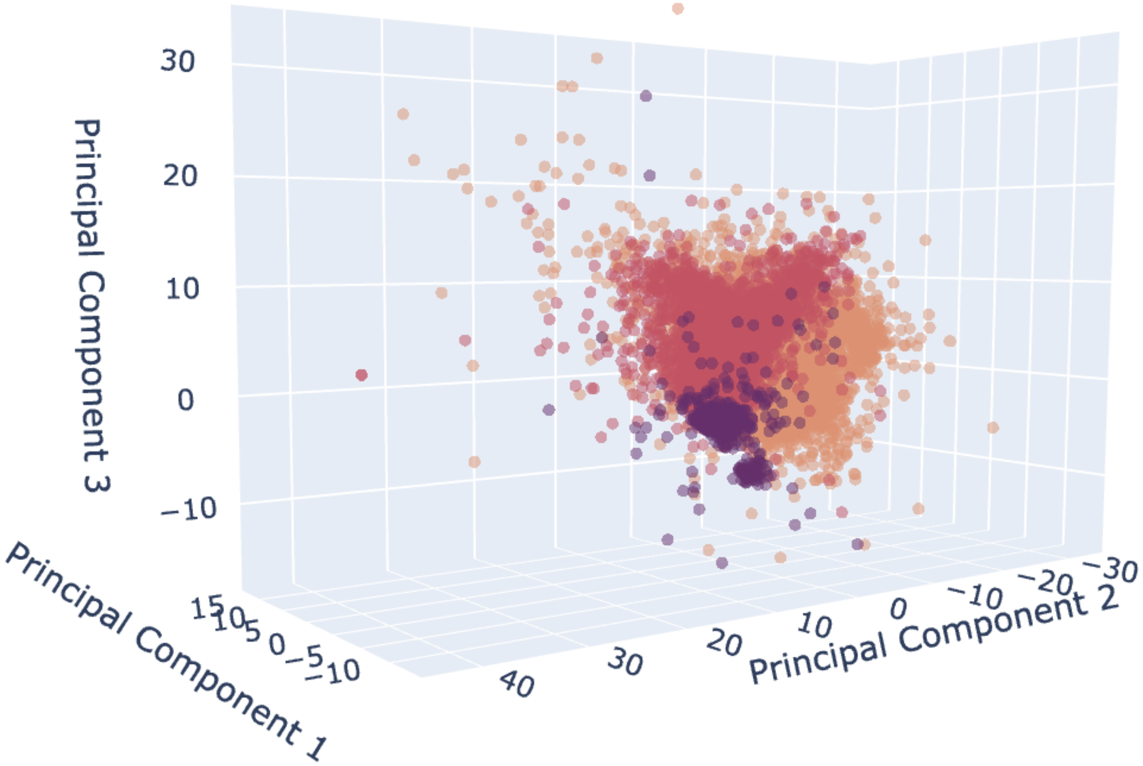

Figure 4: PCA on Original Vectorized Genomes

distinguished clusters seen in the PCA of the latent representations. We also trained our LightGBM model on these input genomes and it had a testing accuracy of 70.02% alpha, 74.19% delta, and 86.23% omicron. Therefore, the original genomes themselves are not that easily separable. With this, we can conclude that our autoencoder is successfully reducing the dimensionality of the input vectorized genomes and allowing for greater separation with the latent representations.

Lastly, we will compare our method of classifying viruses via LightGBM with other existing literature. Specifically, we will compare our LightGBM results with the methods described by Randhawa et. al[2]. They trained and tested six different models with the Quadratic SVM performing the best at 94.9% accuracy (our average accuracy was 88%). However, there task was for taxonomic classification of COVID-19 more so than variant detection, hence their classification criteria was much different. Nevertheless, they also concluded that their "alignment-free whole-genome machine-learning approach can provide a reliable real-time options"[2]. This gives us confidence that our approach can also have similar positive conclusions.

References

[1] S. Elbe and G. Buckland-Merrett. Data, disease and diplomacy: GISAID's innovative contribution to global health. *Global Challenges*, 1:33–46, 2017.

[2] Soltysiak M. El Roz H. de Souza C. Hill K. A. Kari L. Randhawa, G. S. Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *PloS one*, 1:10, 2020.

[3] J. Ren, K. Song, C. Deng, N. A. Ahlgren, J. A. Furhman, Y. Li, X. Xie, R. Poplin, and F. Sun. Identifying Viruses from Metagenomic Data Using Deep Learning. *Quantitative Biology*, 8:64–77, Oct 2019.

[4] Y. Shu and J. McCauley. GISAID: from vision to reality. *EuroSurveillance*, 22:13, 2017.