Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Machine Learning in Bioinformatics

Application of Gated Recurrent Units for Genome Sequence Recognition

Esteban Segarra Martinez [1,*]

[1] Computer Science Department, University of Central Florida, Orlando, 27000, Florida, United States

* estebansegarra@knights.ucf.edu

Associate Editor: No Editor

Received on Fall of 2021

Abstract

Motivation: To apply or extend the paper initially developed by Randhawa in being able to sync or detect the sequence of COVID19 through an alternate method using an gated recurrent unit deep-learning strategy. This paper evaluates the ability of GRUs to locate potential sequences and similar sequences that happen to contain COVID-19 derived sequences. Evaluation will also be applied to see if modifying the DeepGRU strategy being employed will improve the results of the algorithm or hamper them. Afterwards, this paper will evaluate the ability of the DeepGRU gesture recognizer in comparative genomics by comparing the results to the original paper by Randhawa. Results show that while DeepGRU is excellent at learning the patterns of the data even if a sequence is passed for a semi-relevant virus sequence, it cannot reliably determine sequence inheritance such as COVID-19 from a parent source.
Results: Available upon request.
Availability: Send an email to obtain results or read the paper below.
Contact: estebansegarra@knights.ucf.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The novel coronavirus COVID-19 has been the front of research in curving and stopping the pandemic and its harmful effects on infected individuals. Part of this effort has been the ability to detect and perceive it in infected individuals. Aside from detecting it, researchers have also been trying to understand how has the COVID-19 DNA sequence evolved ever since its initial sequencing. Various variants have been discovered as reporting continues on new COVID-19 cases. Researchers are concerned that with evolving cases of COIVD-19, resistances to drugs and previously known remedies may become ineffective. As a result scientists are developing techniques to combat it as well as trace the initial origin of the virus. Deep learning application for pattern recognition of novel or new variants of viruses is not a new problem. Techniques for genome sequencing include the application of genomics tracing software, or deep learning strategies such as in Randhawa's[18] paper Machine Learning with Digital Signal Processing (MLDSP)[17] was used to find variants of COVID through the National Center for Biotechnology Information (NCBI) database. Others are more robustly optimized and widely used such as Illumina, SMRT, and MiniON [4], all commercially available software used for the purpose of DNA sequence variants. These software packages rely on internalized algorithms and databases that can detect comparative sequences to detect similar strands. Despite the availability of this software and due to the magnitude of scale of data that goes into a DNA strand, sequence variants are difficult to detect due to the minuscule mutations that could happen between DNA sequences.

2 Related Works

2.1 Comparative Genomics

As such, toolsets and techniques have been developed to detect changes between sequences. Some of these include TreeWAS [6], CCTSWEEP/ VENN [9], Scoary [5], BugWAS[8], ROADTRIPS[20], SEET[12], PySEER[11], GEMMA[21], FAST-LMM[13], HAWK[16], DBGWAS[10], PLINK[15], Phenotype Seeker[2], Kover[7], and Hogwash[19]. These toolsets all work to respectively for comparative genomics toolsets. Despite this, a toolset or technique that could determine a genome sequence similarity on-the-fly with 100% accuracy is preferred as shown by [18, 1]. In recent years, new machine learning techniques such as those developed for gesture recognition or image recognition show to be promising when applied to other fields outside their developed purpose.

In specific, the understanding and usage of encoding techniques, gated recurrent units, and combining them with convolutional neural networks with work shown such as Maghoumi[14].

2.2 Application of Gated Recurrent Units in Comparative Genomeics

There is relatively recent research in the application of gated recurrent unit for comparative genomics. One such paper uses GRUs to determine transport proteins with a GRU architecture [**?**]. Another paper by the same author also used GRUs to determine the fertility of individuals [**?**]. Another study looked at using GRUs for predicting RNA binding areas [**?**]. Afterwards, research trends to other case uses for GRUs, such as for DNA binding prediction [**?**]. These papers show evidence to the application of GRUs for learning patterns of sequences and locating specific locations of interest should the GRU be setup to do so.

3 Contributions

This paper will look into applying the DeepGRU algorithm as developed and stated in the paper [14]. Prior work shows some successes in using GRUs for sequence detection, and in this paper we would like to see how an optimized version of a GRU used for gesture recognition would perform when applied to genome sequence with the data organized in a trainable manner. Key contributions will include:

- DeepGRU's ability to perform sequence to sequence comparison and matching
- Determination of different sequence species when trained
- Performance improvements when the algorithm is changed to improve the accuracy or improve the algorithm's ability to recognize sequences

As the paper will explore, there are some functions that DeepGRU is appropriate to perform such as determination of a genome between different classes. However it is poor or unable to determine if two sequences are from the same class of species, such as a genetic connection between Riboviria and COVID-19.

4 Approach

The paper initially proposed by Randhawa uses the approach of finding relevant or parallel genomes based on training an gated recurrent unit system on a baseline of sequenced COVID19 to find other sequences. This paper involved the use of the MLDSP algorithm which applied its own processing to develop an understanding of the data using DSP techniques to determine if sequences were similar or not [17]. The basic principles behind MLDSP is that the data is splint into separate sequences and converted into discrete numeric representations. The data is then passed into a discrete Fourier transform in order to determine a magnitude of influence of the sequence. The next step is to create a Pearson correlation coefficient (PCC) that is determined for each magnitude. The PCCs are then passed into a supervised machine learning technique and classified with a 10-fold cross-validation technique.

This paper will utilize a different approach for detecting sequences through an encoder using GRUs in combination of a CNN and attention module as discussed by Maghoumi[14],DeepGRU. The original paper discussed the use of DeepGRU as a technique for detecting and matching gestures and performed in trials with near 100% accuracy when passed with recognizing images with gestures. This is due to the ability of GRUs to compress information into smaller chunks and passing the information into a CNN for recognition.

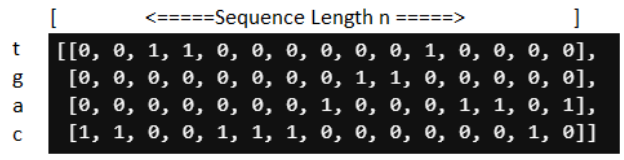

Fig. 1. One-hotting of a sequence of DNA through nucleotides.

4.0.1 Application of DeepGRU to Genome Sequencing

Since comparative genomeics deals with having a big-data problem, the data must be split up in a manner that can be used for best results. To do this, the four types of nucleotides are one-hotted in 4 parts, each array representing one of the nucleotides where 1 represents a nucleotide's presence and 0 representing no presence. After the data is one-hot encoded, the data is then processed to created a so-called masked version of the data. This is done for a later step involving validation and testing, where this muddled/masked data can be used to further verify the training process performed correctly.

4.0.2 Classification

After one-hotting, the data is directly passed into training where the data can be compressed by GRU and trained by CNN. The CNN applied was the same as used in the Maghoumi paper where there are 256 nodes of CNN for two layers. The final layer is a soft-max layer that classify items. The amount of epochs used to train the CNN is case-dependent, but due to the GRU, the amount of epochs required for full training is between 10 - 30 trails to reach a full convergence. Preliminary trails converge to a high classification accuracy after only 5 - 10 training and validation epochs.

4.0.3 Data Selection and Sourcing

The data was sources from the NCBI website [3] and looking at the results gotten from looking at the keyword "Severe acute respiratory syndrome coronavirus 2". As of November 17, 2021, there were 92 recognized COVID-19 variants. With this data source, 15 sequences were chosen at present to serve as a medium of training data. Another 15 sequences were chosen as a validation set and 5 more chosen as testing. Initial training and validation showed promising results as validation results showed a high degrees of matching on between 1 and 5 validation samples.

5 Methods

Following the experiment method originally developed by Randhawa[18], the experiment was designed to use part of the dataset used in that paper. The full dataset wasn't used due to time and resource limitations, however figuring that part of the experiments ran by Randhawa can be replicated by this paper.

5.1 Dataset Selected

As stated in 2.8.2, the dataset selected is the same as in the original paper with some limitations. The goal of the paper was to demonstrate the capability of DeepGRU to be applied as a comparative genomics tool and observe its ability to perform the goal of detecting and determining if a genetic sequence can be determined to be a strain or not from a trained DNA sequence. In this case the strain being tested is COVID-19 with some of its variants being validated on sequences that could be variants or not. In total, there are 29903 individual nucleotides in a complete COVID-19 sequence.

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

5.2 Evaluation and Metrics

Evaluation of the technique applied consists of utilizing the built-in tools in Pytorch for evaluating the accuracy of the technique applied. Most of this was already setup for use by the developers of DeepGRU but could be easily adapted for other purposes. The accuracy metrics used were those used provided by the Pytorch implementation of DeepGRU, which is the accuracy of a trained data sequence for one genome sequence and a validating one. Because DeepGRU was designed to take up time-sequenced data, the transformation of the data then allows a comparison between different streams of data and accuracy would be determined by how many sub-sequences of a genome can be matched together.

5.3 Materials

The experimenter had access to a large-scale desktop computer with 64GB of RAM, a 10850k i9 Intel CPU, and a 2080 with 8GB of VRAM. The datasets were found through looking up the second paper by Randhawa for DSP [17], which contains a repository with the original code for the MLDSP and all of the sequences used for the six tests in the first paper by Randhawa [18]. Additionally to this, I separately downloaded 11 sequences for COVID-19 from the NCBI database and added them to a separate file that can be used for training and testing.

For DeepGRU, I downloaded the repository from the original paper from GitHub on the link https://github.com/Maghoumi/DeepGRU. The original Randhawa paper had used 23 sequences, however because the experimenter had limited resources, adding more sequences would increase a chance for a crash in the code.

6 Experiments

The experiments based on this paper are based on the seven tests done in Randhawa's paper. The seven experiments in this paper use the same datasets as in the experiment with some exceptions. The first four experiments, Tests 1 - 3b use a shortened amount of virus sequences than how the paper presented it. The reason for this is for performance problems. Computationally, DeepGRU requires a lot of resources preloaded into RAM, especially after the sequences are one-hot encoded, this process generates a lot of data that has to accumulate into RAM.

The limitations of the experimenter's capability in the form of VRAM capacity, as once an experiment batch runs out of VRAM at any time, CUDA fails to run and the experiment could freeze during runtime - losing results. As a result, the amount of epochs had to be watched when more than 100 sequences were provided. If the amount of epochs exceeded 100 sequences, the amount of epochs selected would be 25 epochs, with experimentation to see if more epochs could be presented to improve performance. For dataset experiments where there is less than 100 sequences, the amount of epochs used will be 100.

These rules were performed for the last three tests, tests 4 - 6 of which, the results will be shown in the results section of this paper. A successful execution of an test would mean that the dataset is loaded into DeepGRU with unique samples from a training/test/validation split and finishes all the assigned epochs to the end. Sometimes this might fail at random periods but generally keeping the data inputted into the VRAM to be less than the total helps mitigate potential crashes.

The reason for different splits in data was because some experiments require less testing data to verify, for example, if we are trying to see if we can correctly classify COVID-19 results from a dataset that was trained.

6.1 Determining Sequences as Variants

Determining a variant is done by training a sequence on COVID-19 and then passing a potential variant by passing it as a validation or testing variant. Accuracy means that there is a percentage of matching certain parts of the DNA to different sequences. While DeepGRU wasn't trained on determining if individual segments match to a tested variant, the algorithm's accuracy can tell us how similar is one variant to another. This is because accuracy would be reflected as how likely is a testing sequence matches to the patterns of a trained sequences. Higher accuracy means there is a high likelihood of matching the pattern while less accuracy reflects a lesser chance of being a derivation of a trained sequence.

To test this part, DeepGRU was trained per individual sequence and then tested on COVID-19. This allows us to generate accuracy scores per type of sequence (such as betaviruses, alphaviruses, etc.) and observe if there is any correlation between different sequences, allowing an easy way of separating between datasets rather than Randhawa's testing approach.

This part of the experiment will be performed on the Test 1 dataset which has 11 variants of viruses. Randhawa's paper was testing for COVID-19 with sequence with accession number MN908947, and the experiment replicates this by choosing this sequence to test on.

6.2 Modifying the DeepGRU Algorithm

After evaluating the performance with just the default version of DeepGRU, additional investigations can be performed by modifying the DeepGRU algorithm. One of these components includes altering the CNN to have more or less layers to observe performance. Altering the batch size as it goes into the model to be trained can also be used to understand if for DNA sequencing, the amount of data being used to be put into the model affects the performance and accuracy.

Alterations will include; changing the amount of nodes inside the CNN, modifying the batch size, modifying by adding more layers, and changing dropout rate to a rate less than 50%. The dataset that will be investigated will be for the dataset of Test 5, which has four viruses and one more for COVID-19. For testing, COVID-19 variants can be organized into a set for testing. The split between the datasets was 12 instances of COVID-19 for training, 22 for Sarvecovirus, and for validation, 11 instances for COVID-19 and 21 variants of Sarvecovirus.

The specified modifications are being done in order to trial-and-error and see what modifications would improve the overall performance of the DeepGRU. These are the most common changes that can be done to improve the model without significantly changing the architecture of the model. Additional alterations to the model would require a larger investigation which is outside of the scope of this paper.

7 Results

Results from the DeepGRU implementation will be shown here. All of the tests were detailed in the Experiments section. Results below are split between the Randhawa's paper and also a second implementation to attempt to determine similarity by accuracy of sequence. The following section details the results of DeepGRU on the testing datasets of Randhawa's paper.

7.1 Results for Tests 1 - 3

This test consisted of applying all the sequences within evaluation and observe accuracy results. This section was ran in a similar manner as in tests 4 - 6, as there was difficulty in trying to run the DeepGRU to allow a direct sequence detection, it was decided to run these datasets to locate potential COVID-19 mixed in.

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

Test Number	Training Acc.	Validation Acc.	Testing Acc.	Epochs
Test 1	96.97%	55.98%	99.25%	25
Test 2	88.12%	52.77%	99.66%	25
Test 3a	98.71%	79.75%	99.11%	25
Test 3b	98.97%	79.82%	98.97%	25

Table 1. Accuracy for the first three tests of Randhawa's paper tested against the COVID-19 with Accession Number MN908947

Test Number	Training Acc.	Validation Acc.	Testing Acc.	Epochs
Test 4	99.17%	89.09%	86.79%	50
Test 5	89.09%	95.14%	83.13%	25
Test 6	85.85%	87.61%	99.76%	25

Table 2. Test scoring for the last three tests in the Randhawa paper

Training Set Name	Testing Acc	Number of COVID-19 Samples
Riboviria	0.003%	1
Riboviria	60.75%	5
Riboviria	36.60%	9

Table 3. Results of accuracy or in accurately locating a variant of COVID-19 with Accession Number MN908947

7.2 Results for Tests 4 - 6

Tests 4 - 6 results show that the COVID-19 variant was able to be distinguished between the different sets trained upon. The accuracy is shown in the following table:

Results show that the validation accuracy was overall lower than the training or testing datasets. Testing was performed on a single, random variant of COVID-19. Training was included since the DeepGRU didn't complete with 100% or perfect training. Test 6 was tested using the last two COVID-19 variants in my dataset, which had accession numbers MN996531 and MN996530, as the original COVID-19 MN908947 was dropped into the training set as in the original paper. An additional 11 samples of COVID were dropped in between training and validation sets respectively. Different

7.3 Determining Variants

While attempted to run the experiment and preloading the DeepGRU with a specific dataset from each dataset between the 1 - 3 Tests, the capability of detecting a COVID-19 variant was not possible to do so with just the samples taken from the dataset. Doing so without any COVID-19 variant used as a side loaded variant was always 0%, showing that DeepGRU was great at determining different classes of similarly-trained viruses but poor overall if attempting to determine a variant of COVID-19 from a trained set. The only way to perceive a sequence that was inherited from a set was to add to the training set a variant of COVID-19 into the training set,

While the accuracy was non-zero for a training set with 1 single COVID-19 sample, the accuracy was below 0.001.

All of the above results were generated using the original COVID-19 variant as detailed by Randhawa's paper,

7.4 Modification of the DeepGRU

This section details what happens when modifying the DeepGRU algorithm and its results in accuracy.

Changes were done to the test 6 dataset since it provided the least margin of variability, the idea is to be able to compare the accuracy of that from Randhawa's paper and the potential outcomes from changing the algorithm's model.

Modification	Validation Acc	Testing Acc	Time (sec)
None	87,61%	99.76%	520
x2 Layer	87.55%	99.98%	528
x2 Nodes	87.62%	99.982%	1052
25% Dropout	87.55%	99.92%	534

Table 4. Results from Modifying the CNN of the DeepGRU (Dataset from Test 6 Used)

8 Direct Comparisons between Results and Original Paper

Because the implementation of DeepGRU favors certain aspects of the paper, I'll discuss those parts first and also discuss why the parts that didn't work, did not. Randhawa's paper discusses two primary comparison points; the first part compares COVID-19 against a potentially relevant sequence group or sub-genomes and the second part is the ability of the algorithm to detect and determine COVID-19 from a mixed dataset of different viruses in a training set. This means that DNA sequences that would be unlabeled and set at the same class level as the trained variant would be detected.

8.1 Comparison of Potentially Locating Sub-Species using DeepGRU (Tests 1 - 3)

This part was the most difficult to find a comparison from. The reason was that the algorithm was not effective at all to determine a comparison with DeepGRU was because it appears that assembled one-hotted sequences trained on the model cannot determine if two different types of genomes are correlatable or not unless COVID-19 is mixed into the training set. This is possibly because DeepGRU senses that patterns of a similar species of viruses can be trained and learned while different genomes which may have significant variations in the sequence, and DeepGRU's accuracy in determining similarity significantly drops off. The tolerance of DeepGRU in determining a comparatively similar sequence only extends to sets that were trained and validated with sequence species of the same type. The evidence to support this is on Table 3, where we can see that the accuracy only starts increasing from 0% when COVID-19 sequences are added into the training set.

Testing was done by choosing only the original sequence of COVID-19 as to see if any of the Riboviria sequences would be detected as a potential basis. The accuracy metric was chosen as a method to determine if DeepGRU was sensing something rather than loss. Loss did decrease for some sequence types to be closer in similarity, however this was unreliable. It is suspected that the difference in accuracy scores between the five and nine sample sets was that the set with five COVID-19 variants was smaller and more diverse than the set with 9 elements, which could have influenced negatively the results of the algorithm.

8.2 Comparison of Accuracy in Locating Sequences in Datasets (Tests 4 - 6)

The comparison in accuracy for distinguishing between different datasets is the strength of DeepGRU. Assuming different classes are given to each type

Test Number	Testing Acc.
Test 4	98.4%
Test 5	98.7%
Test 6	100%

Table 5. Test scoring for the Randhawa paper (Best scores as seen in Table 3 of paper [18])

of virus type, DeepGRU can effectively distinguish with approximately 80% capability. This compares with the tests 4 - 6 that are presented in Randhawa's paper, however the DSP strategy appears to be stronger in determining a comparison. Randhawa's paper used different algorithms that were used classify the elements in the datasets being trained. The difference for DeepGRU is that it has a CNN built-in directly to create a classification model, which determines accuracy scores without changing the classifier.

The only test that appears to have had a reliably higher score was Test-6, however Randhawa's paper still had a higher score at 100% accuracy. This means that DeepGRU gets confused between the two classes in that dataset due to similarity at the start of the sequences and mis-classifies it. This is congruent in behavior as in the previous section 7.1, where locating sub-species sequences was hard or impossible to do so without having a trained sequence to test against.

8.2.1 Comparison of Accuracy After Modifying DeepGRU's CNN

After modifying the CNN of the DeepGRU, performance was analyzed again for variations of the CNN. Overall behavior was equivalent to the stock, unmodified DeepGRU as provided by the repository. The accuracy seemed to improve greatest when additional layers were used in the classifier model, by duplicating the neural network's, accuracy increased about .20%. However the validation score didn't improve, and it seems that for all other datasets, this probelm remained recurrent and at the same level overall for all of the cases where I modified the CNN. Scores still do not reach 100% accuracy for the equivalent case in Randhawa's paper, however since 99.98% is close enough to 100%, we can say that the model produced near-equivalent accuracy to the original paper.

9 Conclusion

There are different conclusions that can be arrived from the results achieved. For one case we have to observe the tests that were done for distinct reasons; some tests attempted to determine if a target sequence was in the set that was trained on. Other tests were used to determine if a sequence could be located within some margin of accuracy. Despite the accuracy of prediction being that of passing a sequence of DNA through the model, conclusions can be arrived about how DeepGRU is working and how could it be best applied in practice.

9.1 DeepGRU as a Sequence Determinant

Due to time limitations, the current implementation is a bit limited in how it operates in detecting and determining a tested sequence. It is a non-trivial problem to be able to determine if a sequence is within a target dataset, however this might not be the most ideal of cases when directly attempting to compare it against Randhawa's paper, whose DSP algorithm converts the sequences into measurable and scaled variations that can be used to track the different segments of a DNA sequence. This current implementation however can determine if a sequence belongs or not to the currently trained set, determinable by the final accuracy of a sequence that was predicted,

In this sense, the DeepGRU algorithm was effective in locating variants of sequences effectively even with large amounts of sequences that were not part of the targeted sequence. As a result if a sequence of an unknown type of virus were passed through a previously trained model for a specific virus, such as Riboviria, would identify subsequent Riboviria sequences through the model.

9.2 Effectivity of Determining a Sequence as a Variant

Since the DeepGRU algorithm can learn the sequences, this application of the algorithm proved to be effective in learning the unique sequences generated. The similarity scores of the algorithms show that the algorithm worked in principle.

Given with the current implementation however, this failed without the inclusion of external COVID-19 sequences that are added to the training set. This means that the current implementation of DeepGRU is good at determining sequences that were trained on, even if the class chosen is mixed to be the same.

9.3 Improving DeepGRU and Routes of Improvement

DeepGRU overall performs decently when given a training set of two items to respond to. In this instance, 23 of COVID-19 variants along with 43 sequences of the virus strain Sarvecovirus. In this aspect the DeepGRU algorithm was good to take the sets provided to train and validate and produce high accuracy scores. However as seen in the results obtained from an unmodified version of the DeepGRU in table 4, results were less than 100%, as obtained by the original paper. Subsequent alterations all increased the time of computation, however they did have an impact on the accuracy.

From the results provided by Table 4, some of the alterations that gave the greatest improvement in accuracy was increasing the amount of layers and increasing the amount of nodes. Changing the dropout rate also led to a minor improvement in the accuracy, but to a lesser degree than changing the layer and node count. The greatest difference between changing the amount of nodes and layers was that increasing the amount of layers had a lesser impact on the computation time than the amount of nodes. By that effect, doubling the amount of nodes doubled the computation time, which means the DeepGRU was overall much less efficient.

Because data sequences are split up into many sub-sequences, there is a possibility that no matter how the model is created, there will always be some portion of the sequences that will be mismatched in order to be classified. As a result, the way to fix this will be to utilize a generative adversarial network that can be used to generate variations of a sequence and train additional samples. By generating additional samples, DeepGRU will be able to recognize variations of sequences and increasing accuracy, potentially up to 100%. The downside to doing this is that sequences could be generated to sequences that do not exist at all and our 100% would falsely be matched to a non-existent sequence.

10 Future Works

Future work on this paper will incorporate elements of sequencing the sequences of the DNA strands so as to per-sequence strand compare each strand between each of the different DNA sequences. Currently the implementation can train on a set of various sequences and generate a model to predict sequences of DNA to be of one set or not. This allows us to pass a sequence of a DNA strand sequence, however we wouldn't be able to determine the similarity of a DNA strand from one part to the next. For this to be possible, we would have to take the present model and re-write it to be able to strand-by-strand compare a prediction and compare against the label. This would involve a major re-write of the DeepGRU algorithm and possibly strand a larger study. Since this was made within a time limit, this decision was left as a task for a future reader or should time allow.

11 Insights and Personal Gains

The insights from this project include being able to learn how to implement advanced machine learning into a specific problem I had not previously encountered. I am also participating in a separate project which involves the use of a variant of DeepGRU for gesture recognition, however the DeepGRU algorithm was modified by a fellow student to be an autoencoder instead of just having the encoding option as in DeepGRU. From my peer I obtained some understanding on how the code works so as to convert it into a genome sequence recognizer, however I implemented all the code myself while modifying DeepGRU. Additionally, I was able to learn more about how to apply machine learning algorithms and how they work, giving me more opportunity to learn how to prepare and process the data into a model. I would have liked to have worked more into this project and perform additional studies into how DeepGRU can be modified to identify per-sequence similarities, however I am satisfied that my initial belief in how DeepGRU works allowed me to develop this project.

Funding

This work was not funded by any organization. This paper was serviced by the paper's author and equipment.

References

[1]Jonathan P Allen, Evan Snitkin, Nathan B Pincus, and Alan R Hauser. Forest and trees: Exploring bacterial virulence with genome-wide association studies and machine learning. *Trends in Microbiology*, 2021.

[2]Erki Aun, Age Brauer, Veljo Kisand, Tanel Tenson, and Maido Remm. A k-mer-based method for the identification of phenotype-associated genomic biomarkers and predicting phenotypes of sequenced bacteria. *PLoS computational biology*, 14(10):e1006434, 2018.

[3]No author. Severe acute respiratory syndrome coronavirus 2. *NCBI National Center for Biotechnology Information*, 2021.

[4]Vikas Bansal and Christina Boucher. Sequencing technologies and analyses: where have we been and where are we going?, 2019.

[5]Ola Brynildsrud, Jon Bohlin, Lonneke Scheffer, and Vegard Eldholm. Rapid scoring of genes in microbial pan-genome-wide association studies with scoary. *Genome biology*, 17(1):1–9, 2016.

[6]Caitlin Collins and Xavier Didelot. A phylogenetic method to perform genome-wide association studies in microbes that accounts for population structure and recombination. *PLoS computational biology*, 14(2):e1005958, 2018.

[7]Alexandre Drouin, Sébastien Giguère, Maxime Déraspe, Mario Marchand, Michael Tyers, Vivian G Loo, Anne-Marie Bourgault, François Laviolette, and Jacques Corbeil. Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. *BMC genomics*, 17(1):1–15, 2016.

[8]Sarah G Earle, Chieh-Hsi Wu, Jane Charlesworth, Nicole Stoesser, N Claire Gordon, Timothy M Walker, Chris CA Spencer, Zamin Iqbal, David A Clifton, Katie L Hopkins, et al. Identifying lineage effects when controlling for population structure improves power in bacterial association studies. *Nature microbiology*, 1(5):1–8, 2016.

[9]Farhat Habib, Andrew D Johnson, Ralf Bundschuh, and Daniel Janies. Large scale genotype–phenotype correlation analysis based on phylogenetic trees. *Bioinformatics*, 23(7):785–788, 2007.

[10]Magali Jaillard, Leandro Lima, Maud Tournoud, Pierre Mahé, Alex Van Belkum, Vincent Lacroix, and Laurent Jacob. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLoS genetics*, 14(11):e1007758, 2018.

[11]John A Lees, Marco Galardini, Stephen D Bentley, Jeffrey N Weiser, and Jukka Corander. Pyseer: a comprehensive tool for microbial pangenome-wide association studies. *Bioinformatics*, 34(24):4310–4312, 2018.

[12]John A Lees, Minna Vehkala, Niko Välimäki, Simon R Harris, Claire Chewapreecha, Nicholas J Croucher, Pekka Marttinen, Mark R Davies, Andrew C Steer, Steven YC Tong, et al. Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes. *Nature communications*, 7(1):1–8, 2016.

[13]Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature methods*, 8(10):833–835, 2011.

[14]Mehran Maghoumi and Joseph J LaViola. Deepgru: Deep gesture recognition utility. In *International Symposium on Visual Computing*, pp. 16–31. Springer, 2019.

[15]Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American journal of human genetics*, 81(3):559–575, 2007.

[16]Atif Rahman, Ingileif Hallgrímsdóttir, Michael Eisen, and Lior Pachter. Association mapping from sequencing reads using k-mers. *Elife*, 7:e32920, 2018.

[17]Gurjit S Randhawa, Kathleen A Hill, and Lila Kari. Ml-dsp: Machine learning with digital signal processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels. *BMC genomics*, 20(1):1–21, 2019.

[18]Gurjit S Randhawa, Maximillian PM Soltysiak, Hadi El Roz, Camila PE de Souza, Kathleen A Hill, and Lila Kari. Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: Covid-19 case study. *Plos one*, 15(4):e0232391, 2020.

[19]Katie Saund and Evan S Snitkin. Hogwash: three methods for genome-wide association studies in bacteria. *Microbial genomics*, 6(11), 2020.

[20]Timothy Thornton and Mary Sara McPeek. Roadtrips: case-control association testing with partially or completely unknown population and pedigree structure. *The American Journal of Human Genetics*, 86(2):172–184, 2010.

[21]Xiang Zhou and Matthew Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nature genetics*, 44(7):821–824, 2012.