Capstone Project, Jeffrey Jeyachandren **Oct. 25, 2020**

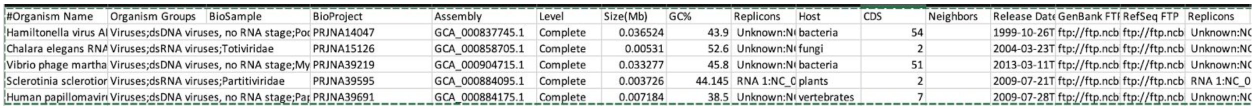
Project Overview

Viruses have many types of hosts. They can infect plants, animals, bacteria, and more. Modern high-throughput DNA profiling techniques have led to an abundance of virus discoveries[1]. However, for most of these discoveries, the virus host remains unclassified. Classification of a virus' host is important for many reasons. It can tell us about the evolutionary relationship between viruses, between viral hosts, and most likely candidates for new hosts. Thus, research into the viral genome has been an important area of healthcare research and computational genomics.

Recently, researchers have had success with using meta-genomic data as an accurate predictor for a viral host[2]. Incorporating features from k-mer compositions and predicted protein domains, scientists were able to greatly improve our ability to classify a virus' host based on simple DNA abstractions. Inspired by this, I would like to evaluate the use of other meta-genomic features such as GC%, CDS, and genome size. These features are defined as follows:

- **GC%**: the percent of guanine or cytosine in DNA. It is highly varied among different types of organisms. Molecularly, it provides stability to the DNA strand and prevents protein denaturing.
- **CDS:** A coding sequence in the DNA. A count of CDS would refer to the number of unique segments in the DNA that start with a starting codon and end with a stopping codon.
- **Genome size**: the size of the sequence measured in mega base pairs (Mb, or 1,000,000 base pairs).

The dataset and inputs for this project come from the Kaggle dataset, "Genome Information for Sequenced Organisms"[3]. Specifically, we will be using the virus.csv file that is provided. The file contains information about 7363 known viruses. For each virus, the dataset has information about it's size (Mb), GC%, replicons, host, CDS, and more. The ML model for this project will be using the viral Size, GC%, and CDS as features to target predictions on the viral host. Here is a sample of the first five entries from the data:



#Organism Name	Organism Groups	BioSample	BioProject	Assembly	Level	Size(Mb)	GC%	Replicons	Host	CDS	Neighbors	Release Date	GenBank FT	RefSeq FTP	Replicons
Hamiltonella virus Al	Viruses;dsDNA viruses, no RNA stage;Poc		PRJNA14047	GCA_000837745.1	Complete	0.036524	43.9	Unknown:N	bacteria	54		1999-10-26T	ftp://ftp.ncb	ftp://ftp.ncb	Unknown:NC
Chalara elegans RNA	Viruses;dsRNA viruses;Totiviridae		PRJNA15126	GCA_000858705.1	Complete	0.00531	52.6	Unknown:N	fungi	2		2004-03-23T	ftp://ftp.ncb	ftp://ftp.ncb	Unknown:NC
Vibrio phage martha	Viruses;dsDNA viruses, no RNA stage;My		PRJNA39219	GCA_000904715.1	Complete	0.033277	45.8	Unknown:N	bacteria	51		2013-03-11T	ftp://ftp.ncb	ftp://ftp.ncb	Unknown:NC
Sclerotinia sclerotior	Viruses;dsRNA viruses;Partitiviridae		PRJNA39595	GCA_000884095.1	Complete	0.003726	44.145	RNA 1:NC_0	plants	2		2009-07-21T	ftp://ftp.ncb	ftp://ftp.ncb	RNA 1:NC_0
Human papillomaviru	Viruses;dsDNA viruses, no RNA stage;Pap		PRJNA39691	GCA_000884175.1	Complete	0.007184	38.5	Unknown:N	vertebrates	7		2009-07-28T	ftp://ftp.ncb	ftp://ftp.ncb	Unknown:NC

[1] Raj, A., Dewar, M., Palacios, G., Rabadan, R., & Wiggins, C. H. (2011). Identifying hosts of families of viruses: a machine learning approach. PloS one, 6(12), e27631. https://doi.org/10.1371/journal.pone.0027631

[2] Lodish H, Berk A, Zipursky SL, et al. Molecular Cell Biology. 4th edition. New York: W. H. Freeman; 2000. Section 6.3, Viruses: Structure, Function, and Uses. Available from: https://www.ncbi.nlm.nih.gov/books/NBK21523/

[3] https://www.kaggle.com/camnugent/genome-information-for-sequenced-organisms?select=viruses.csv

Problem Statement

Currently, only 5% of the viral genomes in the IMG/VR databases are labeled with an associated host[4]. This data is often unavailable since viral DNA samples are taken from the environment and not directly from the hosts. To resolve this, we need to computationally predict and assign viruses to their host to improve the quality of this data set. Unfortunately, most techniques involving virus host prediction are limited to correlations with the host genomes and analysis of k-mers. K-mer analysis is especially problematic since viruses are subject to rapid mutation. Thus, we must look to the field of meta-genomics to help make predictions about viral hosts.

The proposed solution to the problem is to evaluate the predictive potential of meta-genomic data. The meta-genomic data that will be used includes viral genome size, GC%, and CDS count. These features will be used to predict the viral host. They will then be compared to benchmarks that have been established in the scientific literature that use a different set of metagenomic features (k-mer composition and protein domains).

Researchers have most recently used SVM to predict the viral host based on metagenomic features[5]. The accuracy of the model is then evaluated based on the model's prediction of viral hosts given meta-genomic features. To remain consistent with the scientific literature, I will be using the same benchmarks for my model. In previous research, accuracy of 75-100% was acheived. This is the benchmark that I will strive for to make a meaningful contribution to the scientific literature and Kaggle.

Metrics

Since the ML model will be built using SVMs, the evaluation metric will be the predicted accuracy. An accuracy score of 75% or more will be indicative of a predictive signal in the feature set. A score of 0.75 or more will be considered on par with the work that has been done in the scientific literature. Additionally, a confusion matrix of all possible classifications will be used to further evaluate the accuracy of the model. This makes sense for the problem, since a confusion matrix will help us visualize how our model performs for classifications where the viral host infects multiple types of host. For example, predictions for a virus that infects only bacteria must be made differently than a virus that infects both bacteria and humans. These will be separate classifications to be made by the model.

[4] Lodish H, Berk A, Zipursky SL, et al. Molecular Cell Biology. 4th edition. New York: W. H. Freeman; 2000. Section 6.3, Viruses: Structure, Function, and Uses. Available from: https://www.ncbi.nlm.nih.gov/books/NBK21523/

[5] Young F, Rogers S, Robertson DL (2020) Predicting host taxonomic information from viral genomes: A comparison of feature representations. PLOS Computational Biology 16(5): e1007894. https://doi.org/10.1371/journal.pcbi.1007894

Analysis

Data Exploration and Visualizations

After reading in the ‘viruses.csv’ file, the first thing to notice is that the column names contain special characters (spaces, percent sign, etc.) that will be difficult to reference later on. I replaced the column names in the dataframe with the following (note the relevant fields in bold):

- viruses_df.columns = [
'organism_name', 'organism_groups', 'BioSample', 'Bioproject',
'Assembly', 'Level', **'size_mb', 'gc_percent'**, 'replicons', **'host', 'cds',**
'neighbours', 'release_date', 'genbank_ftp', 'refseq_ftp', 'replicons1']

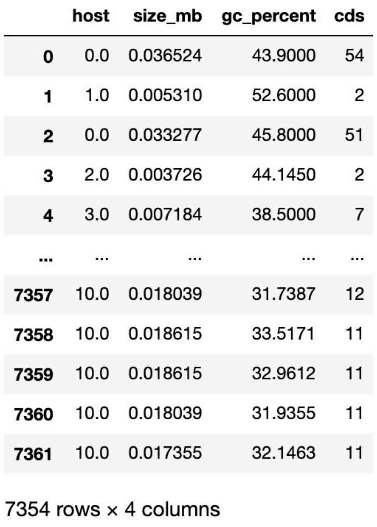
Next, I check on how many unique viruses are in the dataset and how many unique types of hosts there are for the set of viruses:

- Number of viruses: 7362
- Number of unique viral host types: 14
- array(['bacteria', 'fungi', 'plants', 'vertebrates', 'invertebrates',
'protozoa', 'vertebrates, invertebrates, human',
'invertebrates, plants', 'algae', 'vertebrates, invertebrates',
'***vertebrates, human***', 'archaea', ***'human'***, **nan**], dtype=object)

Among the unique viral hosts, there are some viruses that have missing values. This is indicated by the ‘nan’ (bolded above) that is found in the array after printing viruses_df[‘host’].unique() . Additionally, it makes sense to combine the ‘human’ and ‘human, vertebrate’ host types (bolded and italicized above), since humans are vertebrates. If I don’t fix this overlap in the classifications of viral host, it might confuse the model. To distinguish between viruses that infect only humans and viruses that infect humans and other types of hosts, the classification ‘vertebrates, invertebrates, human’ will be used. After dropping the NaN values and combining the vertebrates/human classification, there are 12 unique classifications of a viral host for the model to choose from.

Next, I drop rows with NaN values in the features that are important to us (host, size_mb, gc_percent, cds). I also drop the other columns in the dataset that are not relevant to our problem statement. Finally, I convert the string labels of viral hosts to a more appropriate numerical mapping.

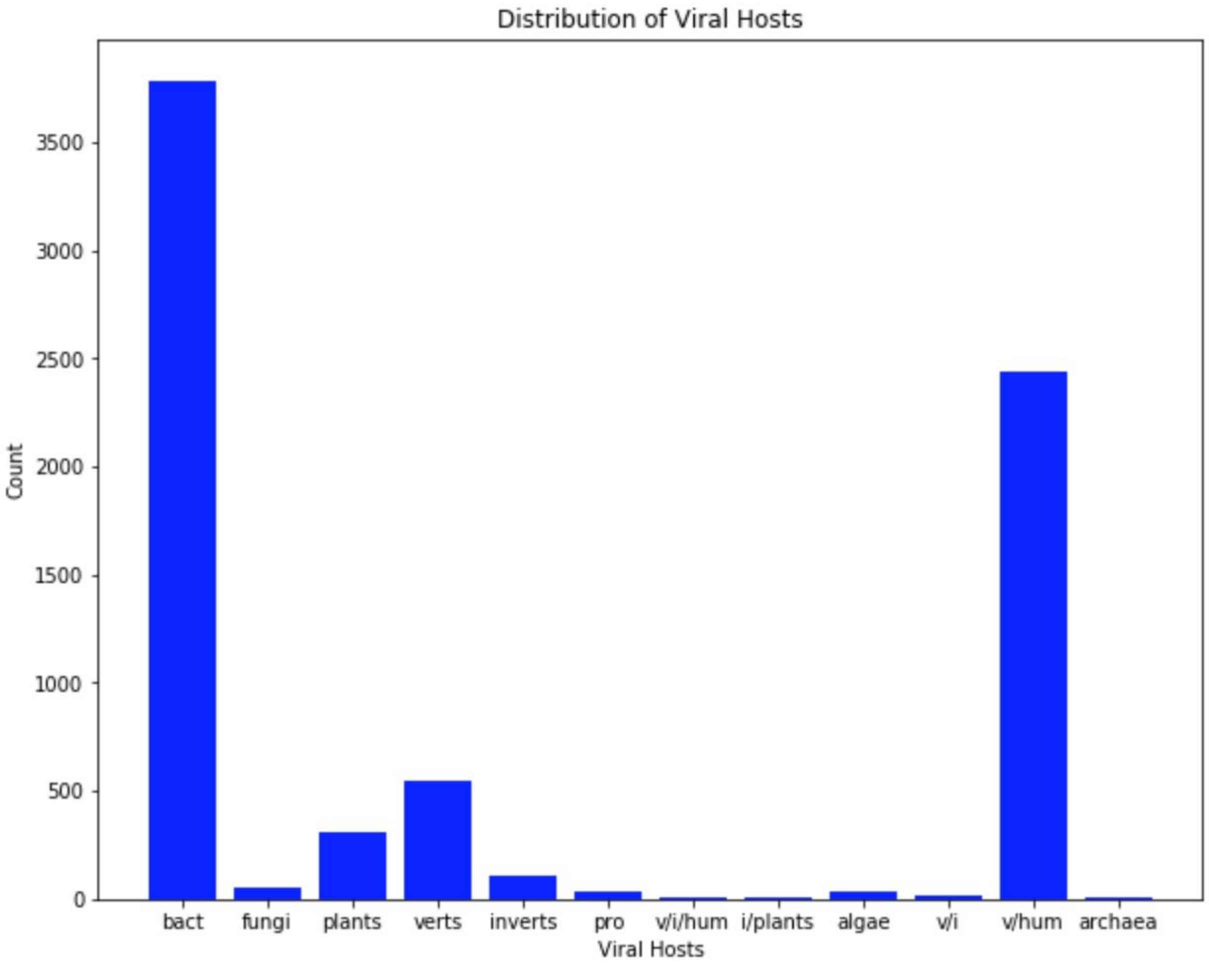
The dataframe now looks like:



	host	size_mb	gc_percent	cds
0	0.0	0.036524	43.9000	54
1	1.0	0.005310	52.6000	2
2	0.0	0.033277	45.8000	51
3	2.0	0.003726	44.1450	2
4	3.0	0.007184	38.5000	7
...	...	...	...	...
7357	10.0	0.018039	31.7387	12
7358	10.0	0.018615	33.5171	11
7359	10.0	0.018615	32.9612	11
7360	10.0	0.018039	31.9355	11
7361	10.0	0.017355	32.1463	11

- 7354 rows × 4 columns

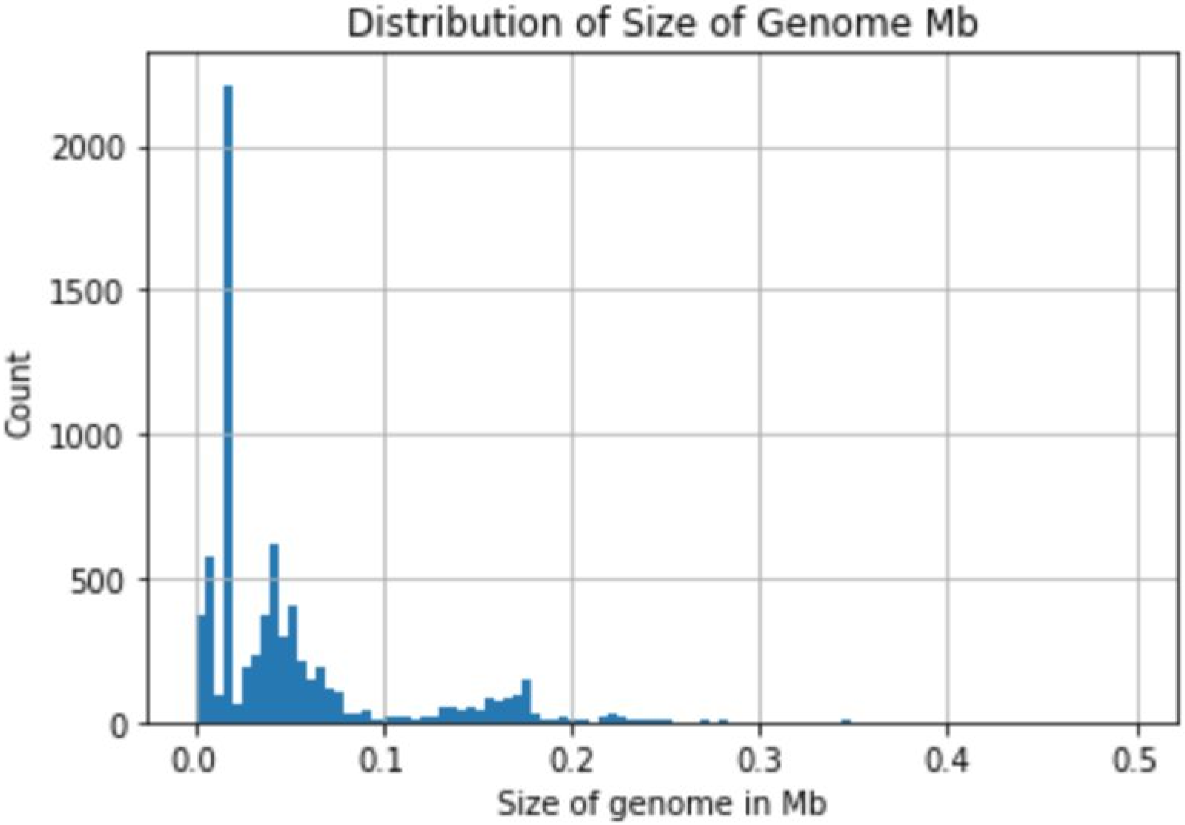
Next, I explore the distribution of the dataset and relevant features. Starting with the count of unique values for each viral host, we see that the data is unevenly distributed between different types of hosts:



There are clearly a lot more bacteria and vertebrate/human viral host classifications in this data set. Later, if the model does not generalize well to the test set, we can use cross-validation to help improve the accuracy (but only if required).

The features that will train the model are the size of the viral genome, the GC%, and the CDS. Starting with the size of the viral genome, I explored some statistics and visualized the distribution of the data in this column:

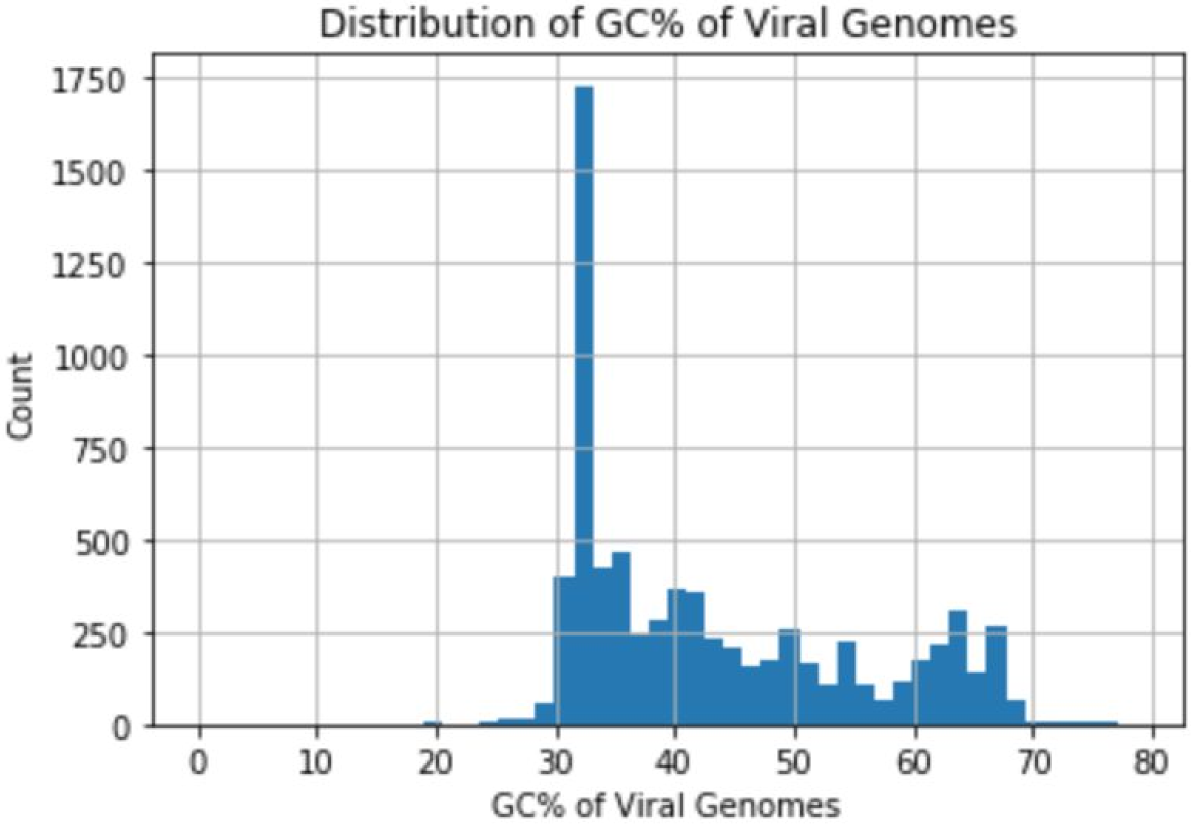
- Number of unique size_mb values: 5233
- Mean of size_mb across dataset: 0.051478 Mb (millions of base pairs)
- Min: 0.000174 Mb
- Max: 0.497513 Mb
- 25% quartile: 0.017989 Mb
- 50% quartile: 0.033620 Mb
- 75% quartile: 0.057982 Mb



As we can see, there is a propensity for viruses to be 0-0.1 Mb in size. The dataset also overrepresented viruses of size ~0.01743 Mb. In the animal kingdom, there are a broad range of sizes for viral genomes. In this case, we might consider using cross-validation if our model does not meet our performance benchmarks.

Similar to how we analyzed the genome size, we will look at the general statistics and distribution of the GC% data for the virus genomes:

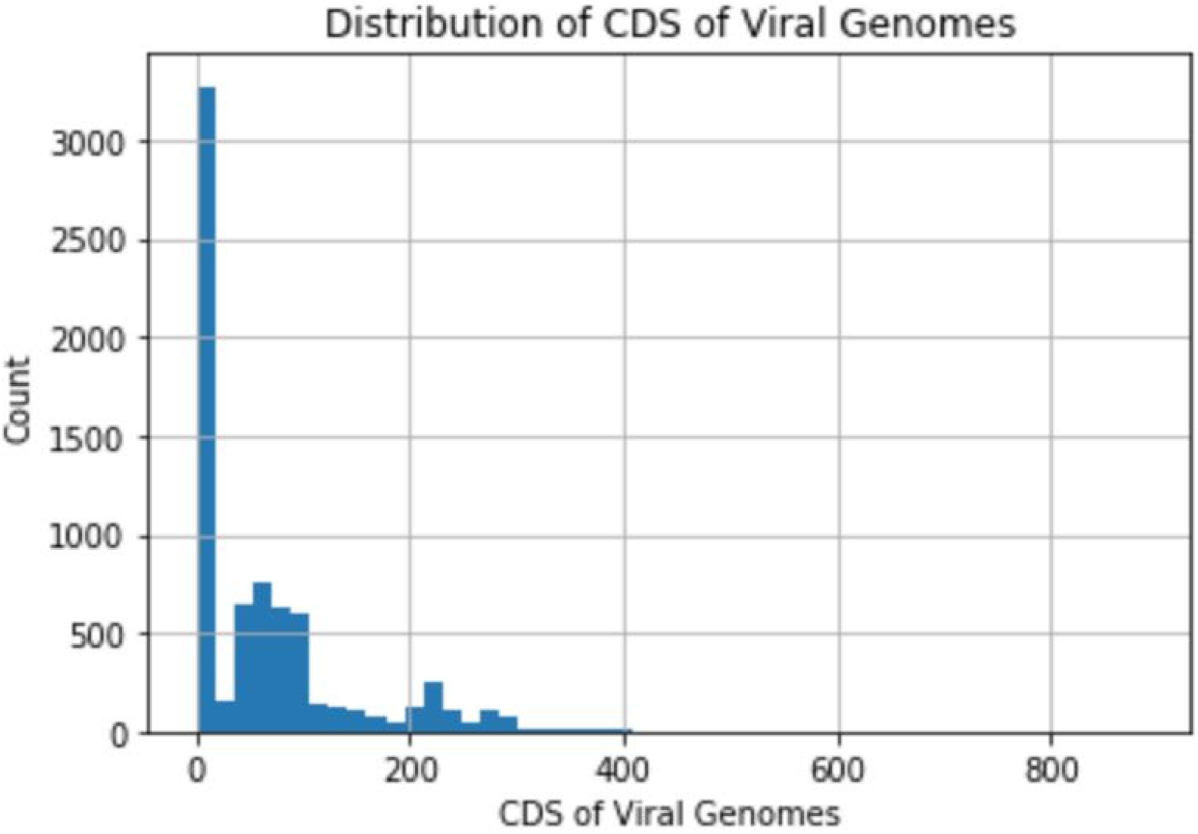
- Number of unique GC% values: 2789
- Mean viral genome GC% across dataset: 43.230327%
- Min: 0.0 %
- Max: 78.8%
- 25% quartile: 32.3709%
- 50% quartile: 39.5%
- 75% quartile: 51.6%



The data for GC% is relatively evenly distributed, with the exception of the GC% of 42.2% being the most common value for this field by far. If this skews the performance of our model, we can use cross validation to correct this.

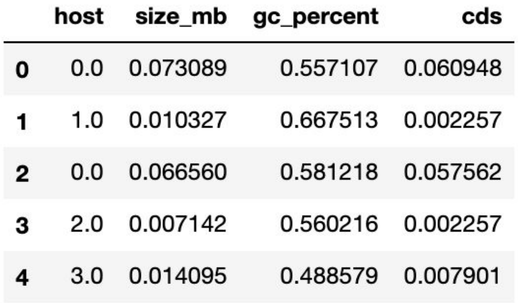
Finally, I explored the basic statistics for CDS among the viral genomes in the data set. The data is summarized and it’s distribution is visualized as follows:

- Number of unique CDS values: 363
- Mean viral genome CDS across dataset: 67.090736
- Min: 0.0
- Max: 886
- 25% quartile: 11
- 50% quartile: 46
- 75% quartile: 90



As can be seen, with only 363 unique values out of 7362 viral genomes, there is the least diversity within the CDS field compared to our other features. The data is also skewed to the left, with many CDS values between 0 and 11 being represented in this dataset (only 32 values of zero). Cross validation is a good solution to address the left-skewing nature of the CDS feature.

Our features capture a range of numerical values and varying units of measurement. For this data to be used, I normalized the values in each feature’s column. I chose to use the MinMax scaler to preprocess these values. This scaler allows us to convert each value to its respective percentage along the range of values. This makes sense since we expect viruses with similar hosts to share similar features along the spectrum of possible values for each feature. For example, viruses that infect vertebrates and humans may share similar sizes or GC%. A combination of these normalized features will help us predict the viral host. After preprocessing the data with the MinMaxScaler (and checking for outliers), our data looks like:



	host	size_mb	gc_percent	cds
0	0.0	0.073089	0.557107	0.060948
1	1.0	0.010327	0.667513	0.002257
2	0.0	0.066560	0.581218	0.057562
3	2.0	0.007142	0.560216	0.002257
4	3.0	0.014095	0.488579	0.007901

Algorithms and Techniques

For this project, I used the SVM algorithm and cross validation to predict the viral host based on meta-genomic features (size of genome in Mb, GC%, and number of coding sequences). SVM was chosen since it is well suited for supervised learning problems. It is also the scientific standard for using ML to predict viral hosts, based on the most recent scientific literature[6]. I evaluated the use of multiple kernels for this project, including linear and rbf kernels for SVMs. The rbf kernel gave an almost 6% improvement to accuracy compared to the linear kernel. Based on these results, rbf was chosen as the kernel for training the model with cross validation.

In the initial training, a train-test-split of 0.8 for training data and 0.2 for testing data was used. However, as noted in the analysis section, there are imbalances in the target data and the feature sets that might skew the results. After evaluating the imbalances in the feature sets, cross validation was determined to be an appropriate technique to resolve these issues. Ten folds were used to validate the data set. The use of cross validation improved the accuracy of the model by 1.5% compared to the initial use of train-test-split.

Benchmark

As mentioned above, Researchers have most recently used SVM to predict the viral host based on metagenomic features[7]. The accuracy of the model is evaluated based on the model's prediction of viral host given meta-genomic features. To remain consistent with the scientific literature, I will be using the same benchmarks for my model. In previous research, the accuracy of predictions ranged from 75% to 100%. This is the benchmark that I will strive for to make a

[6] Young F, Rogers S, Robertson DL (2020) Predicting host taxonomic information from viral genomes: A comparison of feature representations. PLOS Computational Biology 16(5): e1007894. https://doi.org/10.1371/journal.pcbi.1007894

[7] Young F, Rogers S, Robertson DL (2020) Predicting host taxonomic information from viral genomes: A comparison of feature representations. PLOS Computational Biology 16(5): e1007894. https://doi.org/10.1371/journal.pcbi.1007894

meaningful contribution to the scientific literature and Kaggle. I will accept a measured accuracy of 75% or more as a successful evaluation of the unique meta-genomic features chosen to make viral host predictions.

Methodology

Data Preprocessing

The data pre-processing steps occurred as follows:

1. Renamed columns in data set to remove erroneous characters such as white spaces, percent signs, periods, etc.
2. Removed rows that contain missing values. 8 rows were removed that were missing a viral host classification.
3. Removed unused columns. The target column was the viral 'host'. The training features were size_mb, gc_percent, and cds. All other columns were dropped since they only contained meta-data and were not related to meta-genomics.
4. The features were normalized with a MinMax scaler. This was chosen so that I could preserve each item's relative position in the range of values, and also because there were no outliers in the dataset.
5. The host was moved to the first column. This was done so that in the future, the dataframe can be easily converted to a csv format and used in experiments where the classifier expects the target data to be in the first column.

Implementation

After the data was cleaned, the implementation of the classifier occurred as follows:

1. Train-test-split was used with a train_size of 0.8
    a. Later, a cross-validation was used with 10 folds
2. The SVM was initialized with a linear kernel and fitted with the training and test data
    a. Later, a rbf kernel was used in the SVM
3. The linear/rbf classifiers were evaluated by calling predict() with the test set
4. The SciKit Learn score() method was used to evaluate the accuracy of the classifier
    a. Afterwards, the SciKit Learn cross_val_score class was used to evaluated the cross-validation approach
5. A confusion matrix was outputted to further evaluate the results.

Refinement

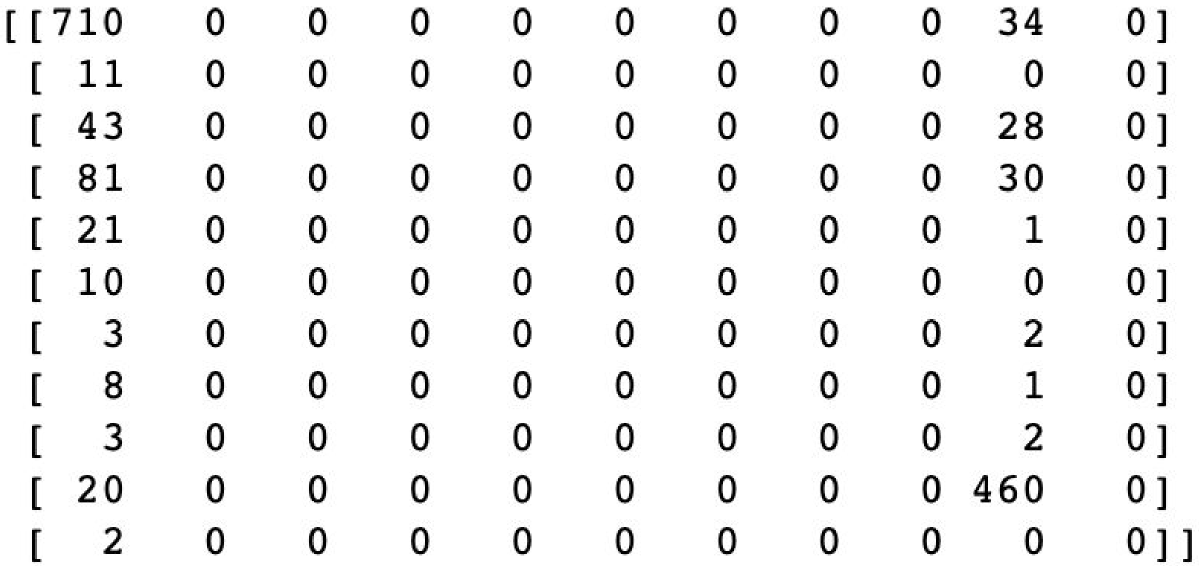
Refinement of the algorithms and techniques proceeded as follows:

- An SVM with linear kernel and an SVM with rbf kernel were independently evaluated.
- Based on the imbalances in the dataset, cross-validation was used and applied to the SVM kernel with higher accuracy.

Model Evaluation and Validation

The accuracy of the model using SVM with a linear kernel was 79%. This initial training already cleared our benchmark from the scientific literature, which was 75%. The model was further improved by using an rbf kernel in the SVM, which boosted the accuracy to 84.5%. Because of the imbalances in the data-set, cross validation was used to further improve the model. Using cross-validation, our model improved another 1.5% to reach a final accuracy of 86%. This is an excellent outcome, and it is great to see such high accuracy despite the imbalances in the dataset. Cross-validation helps validate the model despite these imbalances. A high accuracy with the cross validation technique indicates great robustness for the model.

To further evaluate the model, I looked at the confusion matrix. As expected, the viral hosts targeting bacteria and vertebrates/human had excellent accuracy. This is because these viral hosts were over represented in the data. Classification for other hosts was also generally good. The model performed poorly for plants only (see 3rd row in matrix). This may be due to plants being an overly general label, since it may overlap with some of the other classifiers, including invertebrates, fungi, algae, and archaea.



```
[[710   0   0   0   0   0   0   0   0  34   0]
 [ 11   0   0   0   0   0   0   0   0   0   0]
 [ 43   0   0   0   0   0   0   0   0  28   0]
 [ 81   0   0   0   0   0   0   0   0  30   0]
 [ 21   0   0   0   0   0   0   0   0   1   0]
 [ 10   0   0   0   0   0   0   0   0   0   0]
 [  3   0   0   0   0   0   0   0   0   2   0]
 [  8   0   0   0   0   0   0   0   0   1   0]
 [  3   0   0   0   0   0   0   0   0   2   0]
 [ 20   0   0   0   0   0   0   0   0 460   0]
 [  2   0   0   0   0   0   0   0   0   0   0]]
```

Justification

The final results are excellent and exceed the benchmark expectations. The final model has an accuracy of 86% when using SVM (rbf kernel) with cross-validation and 10 folds. Cross validation ensures the model is robust and generalizable to new meta-genomic data. The model can be used in the future to predict the viral host when this information is not available or missing from viral genome datasets. The model also offers a modest contribution to biological scientific literature, since it indicates that there are predictive signals for viral hosts found in meta-genomic features, specifically the size of a viral genome, the GC%, and the CDS. This might further indicate evolutionary relationships in these features and their significance in viral mutation to new hosts.