COSI 178A
Computational
Molecular Biology

Term Project Report

Predict the Origin of COVID-19 Via Machine Learning

Daihao Xue daihaoxue@brandeis.edu
Junda Li jundali@brandeis.edu
Zhuolin Mao zmao@brandeis.edu

2020.05.12

Table of Contents

1. Introduction

1.1 Background & Motivation

An outbreak of COVID-19 was detected in mainland China in December of 2019. As of this writing, every continent in the world has been affected by this highly contagious disease, with nearly a million cases diagnosed in over 200 countries worldwide. The cause of this outbreak is a new virus, known as the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). On February 12, 2020, WHO officially named the disease caused by the novel coronavirus as Coronavirus Disease 2019 (COVID-19). Coronaviruses are a family of viruses that can cause mild to moderate upper-respiratory tract illnesses such as the common cold, severe acute respiratory syndrome (SARS) and Middle East respiratory syndrome (MERS)[1]. Until the beginning of May, over 4 million cases were confirmed[2].

The debate and studies of the animal origins of COVID-19 has been carrying on. Because it's essential to know who or what infected "patient zero", and tracking down the animal origin can also help with discovering properties and mutations of the virus, investigating how the virus leap from animals to humans, hunting for the next potential coronavirus animal host and getting clues about risk factors for preventing the infections or outbreaks in the future, etc.

1.2 Project Description

Traditionally, most medical scientists will use traditional bioinformatics tools such as BLAST sequence alignments. However, our group would like to analyze the coronavirus genomes from a data science perspective. Specifically, since this is a classification problem, we think it is feasible to use machine learning methods to deduce the origin of the COVID-19, by comparing its genome sequences with other coronaviruses that have been infected on other animals.

Within the machine learning process, we chose the k-mer counting method to do the feature extraction. Then we used Naïve Bayes, KNN and XGBoost models to do the learning and classification. After using the machine learning method, we used Multiple Sequence Alignment tool and compared our machine learning result to the MSA result and got our conjecture.

2. Datasets

2.1 Data Resource

Ideally, we would analyze as many species' all kinds of viruses' nucleotide sequences as possible. But due to limited resources, we used the nucleotide sequences data on NCBI, of coronaviruses that circulate among 5 animals: Chicken, Bat, Camel, Monkey and Pangolin. And we got the genome sequence of SARS-CoV-2 via NCBI's newest coronavirus database.

2.2 Files Description

In the project folder, *human.fasta* is the fasta file of SARS-CoV-2 sequences downloaded from NCBI's Novel Severe acute respiratory syndrome coronavirus 2 genome data hub[3]. *Bat.fasta*, *pangolin.fasta*, *monkey.fasta*, *chicken.fasta* and *camel.fasta* are the fasta files of coronaviridae circulated among bats, pangolins, monkeys, chickens and artiodactyla, respectively, downloaded from NCBI Virus Database[4].

3. Method

3.1 Data Preprocessing

3.1.1 File Processing

We download the sequences of coronaviridae virus of 5 kinds of animals from NCBI Virus and save them as fasta files. And human COVID-19 sequences as a fasta file, too. We want only sequences for our machine learning task so the other information like names starting with '>' in those files should be removed before doing the next step. So while reading those files, we skip lines that begin with '>' and only load sequences to memory. The extracted sequence characters are then appended sequentially into a list. The list contains the whole string for each animal genome.

3.1.2 Three Approaches

There are 3 general approaches for using the downloaded biological sequence data for machine learning. The first way is encoding the sequence information as an ordinal vector and working with that directly. The second way is to one-hot encode the sequence letters and use the resulting array. The third approach is to treat the DNA sequence as a language, a.k.a as text, and use various "language" processing methods. A problem of the first two methods is that they cannot result in vectors of uniform length, which is a requirement for feeding data to a classification or regression algorithm. So we use the last approach: treating DNA sequence as a "language", otherwise known as k-mer counting.

3.1.3 N-gram Extracting

We cannot fit the sequences to machine learning models directly. What we do is to cut the whole sequence into subsequences with length of 4 or 5. This is because A gene sequence should be longer than 4 nucleic acids. We do not need to consider shorter subsequences. For example, a sequence "ATTCGGAT" should be cut as "ATTC", "TTCG", "TCGG", "CGGA" and "GGAT".

Then we need to get numeric representation of those features. We apply the Bag-of-Word model to the dataset and compute the TF-IDF value for each feature.

And finally, we labelled the RNA sequences with their corresponding file name.

3.1.4 Dimension Reduction & Fill NA

Using 4-gram, we got more than 2000 features in our train data. Usually too many features will cause overfitting, which will do harm to our model. Also, they will decrease our training speed and waste a lot of time.

We tried two approaches to reduce the dimension of our data: PCA and counting NaN.

PCA:

Generally speaking, PCA(Principal Component Analysis) tries to find some features that are orthogonal, or in other words, independent of each other in order to reduce those redundant features. To apply PCA, we need:

1 Calculate the covariance matrix *X* of data points: Cx=1n - 1(X-X)(X-X)T, X is a matrix with m lines and n dimensions.

2 Calculate eigenvectors and eigenvalues.

3 Sort eigenvectors in decreasing order according to their eigenvalues.

4 Choose first k vectors and the original matrix will have only k dimensions.

Count NaN :

Sometimes some features only appear in certain species. So after concating all features, we will get many NaN in our train data. So we count the number of NaN in each feature and drop those features with more than 3000 NaN. After that, we fill the NaN with 0. This is because in the BOW model, 0 means absence. So we use 0 to describe the absence of a feature in a sequence.

3.1.5 Oversampling

Our dataset is imbalanced. We have more than 4000 sequences for bats, camels and chickens, but only less than 10 sequences for monkeys and pangolins. If we fit this data to machine learning models, apparently sequences for monkeys and pangolins will be 'ignored' while training. So we also tried two approaches for oversampling: simply copy and paste, SMOTE.

Copy & Paste:

As we keep 4000 sequences for bats, camels and chickens, we tried to simply copy those sequences for monkeys and pangolins for many times until the amount of monkey and pangolin sequences reached 4000. It is simple but we had the balanced data after that anyway.

SMOTE:

Synthetic Minority Oversampling Technique is a method to oversample the minority in a dataset. To apply this algorithm, we need:

1 Randomly choose one item from the minority and use KNN to find its k nearest neighbors.

2 Choose one item from these neighbors(note as b) and put a random point in the line starting with b and original point(note as a):c=a +rand(0,1)*|a-b|

We need to repeat 1 and 2 until the dataset is balanced.

3.2 Model Building

There are a variety of machine learning algorithms to be used. In this project, we explored the following machine learning algorithms.

3.2.1 Naïve Bayes

Naïve Bayes is a classification method based on Bayes rule. It assumes each feature is conditionally independent. It predict the label of each data instance by calculation the probability of p(class | given features) using Bayes rule and pick the class with maximum probability.

3.2.2 KNN

KNN algorithm assumes that similar things exist in close proximity. It labels each data instance by finding its k closest nearby data points and label it with the most common class in the k closets nearby points.

3.2.3 XGBoost

XGBoost is an enhanced gradient boosted decision tree algorithm. Compared to traditional decision tree algorithms such as C.45 or J-48, it improves speed and performance. It is an ensemble tree method that apply the principle of boosting instead of using just one decision tree to predict the data

4. Tests and Results

4.1 Result of Data Preprocessing

As described in the previous part, the original file look like this:

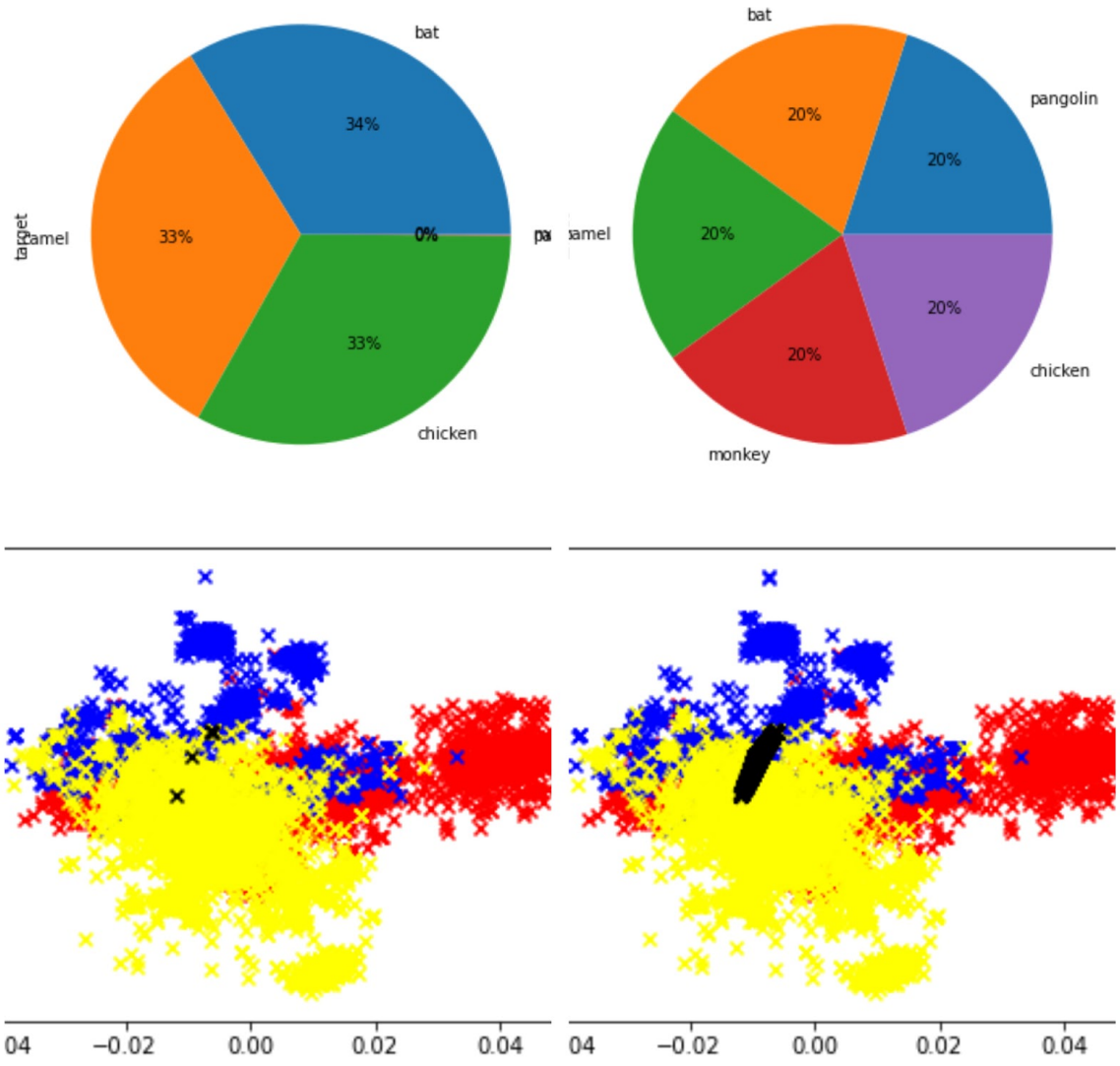
```
>MN183146 |Bat coronavirus isolate 19020 polyprotein| RNA-dependent RNA polymerase region| gene| partial cds
TGGTTGGGACTATCCTAAGTGTGACAGGGCTTTACCTAATATGATTCGTATGATTTCTGC
CATGATTTTAGGCTCAAAACATACGACTTGTTGCGATACTGATGAACGCTATTACAGGTT
GTGTAATGAATTAGCACAAGTGTTGACTGAAGTTGTGTATTCTAATGGTGGTTTTTATCT
TAAACCTGGTGGCACCACATCTGGAGATGCAACCACAGCATACGCTAACTCTGTTTTTAA
CATTTTTCAGGCTGTCAGTGCCAACATCAATAGGTTGTTGAGCATCGATAGTAATACATG
TAATAATGTTAATGTTAAAGAGTTACAGCGAGAATTGTACGATAATTGTTATCGCTCATC
AAGTGTGGATGATGGTTTTGTTGATAAGTATTATAATTATCTGCAAAAACATTTTTCTAT
GATGATTCTATCTGATGATGGA
>MN183147 |Bat coronavirus isolate 19037 polyprotein| RNA-dependent RNA polymerase region| gene| partial cds
TGGTTGGGACTATCCTAAGTGTGACAGGGCTTTACCTAATATGATTCGTATGATTTCTGC
CATGATTTTAGGCTCAAAACATACGACTTGTTGCGATACTGATGAACGCTATTACAGGTT
GTGTAATGAATTAGCACAAGTGTTGACTGAAGTTGTGTATTCTAATGGTGGTTTTTATCT
TAAACCTGGTGGCACCACATCTGGAGATGCAACCACAGCATACGCTAACTCTGTTTTTAA
CATTTTTCAGGCTGTCAGTGCCAACATCAATAGGTTGTTGAGCATCGATAGTAATACATG
TAATAATGTTAATGTTAAAGAGTTACAGCGAGAATTGTACGATAATTGTTATCGCTCATC
AAGTGTGGATGATGGTTTTGTTGATAAGTATTATAATTATCTGCAAAAACATTTTTCTAT
GATGATTCTATCTGATGATGGA
```

We compared the result of 4-gram and 5-gram, and decided to use 4-gram. After extracting the 4-gram from those files, we have our original train set:

	AAAA	AAAC	AAAG	AAAK	AAAM	AAAN	AAAR	AAAT	AAAY	AACA	...	YWRY	YWWK	YWYW	YYAC	YYCA
5540	0.004427	0.003320	0.002214	NaN	NaN	0.0	0.0	0.004427	0.0	0.003320	...	0.0	0.0	0.0	NaN	0.0
2295	0.000000	0.011869	0.002967	NaN	NaN	0.0	0.0	0.002967	0.0	0.011869	...	0.0	0.0	0.0	NaN	0.0
3159	0.002633	0.003292	0.004608	NaN	NaN	0.0	0.0	0.005925	0.0	0.005925	...	0.0	0.0	0.0	NaN	0.0
6892	0.008671	0.002890	0.005780	NaN	NaN	0.0	0.0	0.005780	0.0	0.005780	...	0.0	0.0	0.0	NaN	0.0
2156	0.002825	0.008475	0.005650	NaN	NaN	0.0	0.0	0.000000	0.0	0.005650	...	0.0	0.0	0.0	NaN	0.0
4005	0.008571	0.002857	0.005714	NaN	NaN	0.0	0.0	0.005714	0.0	0.000000	...	0.0	0.0	0.0	NaN	0.0
4807	0.002469	0.003086	0.004321	NaN	NaN	0.0	0.0	0.004938	0.0	0.006790	...	0.0	0.0	0.0	NaN	0.0
765	0.000000	0.016194	0.000000	NaN	NaN	0.0	0.0	0.004049	0.0	0.008097	...	0.0	0.0	0.0	NaN	0.0
2464	0.002488	0.003731	0.003109	NaN	NaN	0.0	0.0	0.004353	0.0	0.006219	...	0.0	0.0	0.0	NaN	0.0

At this time, our data is still imbalanced. We do not want to lose the feature name(subsequences), and it is hard to decide how many features we should keep, so we decide not to use PCA and simply set a threshold as 9000 to drop features.

And we decide to use SMOTE to do the oversampling and we have some comparisons between the data before SMOTE and after:



Above is the data distribution pie chart and data distribution showed after reducing dimension to 2 using PCA. From both images, we can see that our SMOTE oversampling is working well. We can see from the second comparison that all the 'bat' sequences cluster in a small area. We can infer that those sequences should not be like this from the distributions of sequences with other labels. The best way to solve this problem is looking for some new bat sequences from websites but we cannot find other resources for more sequences, so this is the best way we can do.

4.2 Result of the Model

We split the whole dataset and use 80% to train and 20% to do the validation. We first ran Naïve Bayes and conducted model tuning on parameter alpha to find the best model. The result is shown below.

Naïve Bayes		
	No tuning	After tuning
F1-Score	0.73	0.77
Accuracy	0.72	0.78

We also ran a baseline dummy classification by assigning the most frequent class for all data instances. The result for the dummy classifier is 0.20.

The Naïve Bayes work much better than dummy classifiers. However, the above table suggests that tuning on Naïve Bayes did not produce significant improvement on accuracy. As we can see from the above table, the accuracy only raised by 6 % to 0.78.

In comparison, the XGBoost and KNN work much better. From the table below, the XGBoost and KNN both could achieve an accuracy of 0.99 without model tuning. This suggests XGBoost and KNN is a more suitable method.

	Naïve Bayes	XGBoost	KNN
F1-Score	0.77	0.99	0.99
Accuracy	0.78	0.99	0.99

As the performance of XGBoost and KNN is much better than Naïve Bayes model for our dataset, we choose to use XGBoost (Learning Rate = 0.3) and KNN(k = 5) to predict the final result, which is the origin of human COVID-19. And the result is listed in the table below:

XGBoost:

Bat	Camel	Chicken	Monkey	**Pangolin**	TOTAL
95	17	18	0	**1073**	1203

KNN:

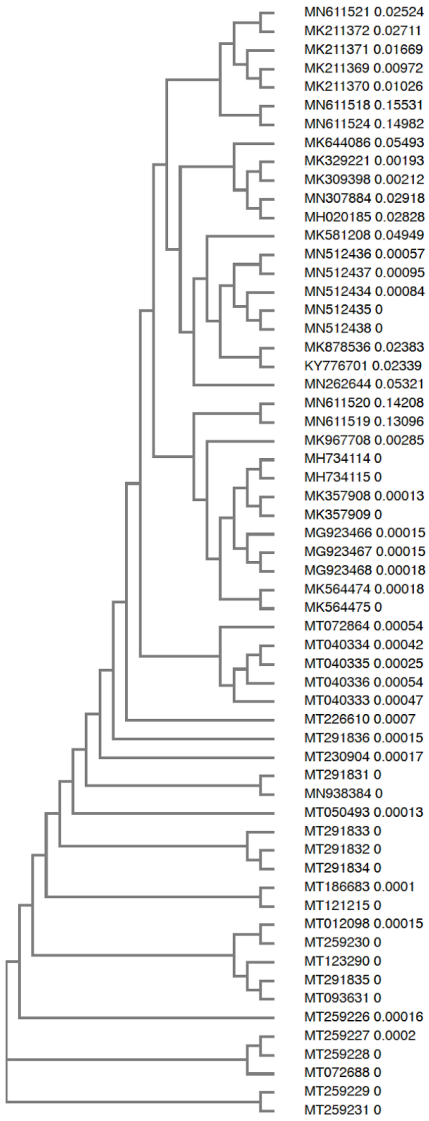
Bat	Camel	Chicken	Monkey	Pangolin	TOTAL
1169	20	0	0	14	1203

According to the result, most of the human COVID-19 sequences are predicted as ‘Pangolin’ or ‘Bat’. Since KNN and XGBoost give different answers, we could decide which is the actual origin.

So our conclusion is: according to our models, human COVID-19 virus should come from pangolins or bats.

4.3 Comparing with the Result of Multiple Sequence Alignment

We used Multiple Sequence Alignment tool Clustal Omega on EMBL-EBI[5]. Due to limited tool capacity, we filtered out 22 sequences of human SARS-CoV-2 and 38 coronaviridae of the five-animal species we chose to be the sampled data to do the alignment. The filtering is based on collection year and completeness. The alignment result is in the file *Multiple Sequence Alignment-CLUSTAL.txt*. We used Jalview[6] to see the visualization and analysis of the multiple sequence alignment. And here is the phylogenetic tree:



From the multiple alignment result we have that the sequences of bats and pangolins coronaviruses have the lowest E values. From the phylogenetic tree, we can see that sequences from MT226610 to the bottom are all human COVID-19 sequences. And from MT072864 to MT040333 are all pangolin coronaviruses, who are probably mutated from the origin (parent node) of all remaining sequences and itself. And that node could come from the parent node of the COVID-19 sequence and itself. So we conjectured that COVID-19 sequences are most aligned with pangolins coronaviruses. So the multiple alignment also shows that COVID-19 could have originated from bats or pangolins coronaviruses, which is in accordance with one of our machine learning results.

5. Discussion

Initially, we predict the origin as bats at first. This is because most of the research and news suspect that the virus comes from bats. However, our work also suggests a different organism, which is Pangolin. Is our result a coincidence or there are also other researchers suggesting the same answer?

So we searched for more research articles to see if there are some other teams reporting the possibility. Some researchers also suspect pangolins may be the origin[7]. However, another team suggests pangolin may not be the origin but could be the intermediate host from bat to human[8]. In addition, a more recently published article[9] argues that it is not likely to come from Pangolin.

It seems that researchers still have different opinions about the origin of COVID-19. Also, in our benchmarks, we also found out that the result is not so stable. They will change along with the change of some hyperparameters of the machine learning model. Nevertheless, no matter how we tune the hyperparameters, the results will always be bats or pangolins, which means these two species are likely to be the origin. The result of the multiple sequence alignment and the phylogenetic tree of sampled data is in accordance with our machine learning result. Although we cannot decide which one is the actual origin host of SARS-CoV-2, our study could also shield light that they are closely related to SARS-CoV-2.

In the future, we should probably try to get some more sequences from different species, and try some more models. However, it should point out that this project is different from normal AI tasks found on Kaggle. This is an exploratory investigation and the correct answer (actual origin of SARS-CoV-2) is unknown. As a result, there is no way for us to evaluate our model, which makes it hard to do the hyperparameter tuning. Maybe one day after there is a conclusion where the virus comes from, we will see whether our model is correct.

References

[1] Carmosino, A., 2020. *Background & History Of The Coronavirus (COVID-19) | Psych Central.* [online] Psych Central. Available at: <https://psychcentral.com/coronavirus/background-history-of-the-coronavirus-covid-19/> [Accessed 1 May 2020].

[2] Our World in Data. 2020. *Coronavirus (COVID-19) Cases - Statistics And Research.* [online] Available at: <https://ourworldindata.org/covid-cases> [Accessed 2 May 2020].

[3] Ncbi.nlm.nih.gov. 2020. *NCBI Virus.* [online] Available at: <https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide&VirusLineage_ss=Severe%20acute%20respiratory%20syndrome%20coronavirus%202,%20taxid:2697049> [Accessed 13 May 2020].

[4] Ncbi.nlm.nih.gov. 2020. *NCBI Virus.* [online] Available at: <https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide&HostLineage_ss=Gallus%20gallus%20(chicken),%20taxid:9031&HostLineage_ss=Pholidota%20(pangolins),%20taxid:9971&HostLineage_ss=Chiroptera%20(bats),%20taxid:9397&HostLineage_ss=Cercopithecidae%20(Old%20World%20monkeys),%20taxid:9527&HostLineage_ss=Artiodactyla%20(whales,%20hippos,%20ruminants,%20pigs,%20camels%20etc.),%20taxid:91561&HostLineage_ss=Homo%20sapiens%20(human),%20taxid:9606&VirusLineage_ss=Coronaviridae,%20taxid:11118> [Accessed 6 May 2020].

[5] Madeira F, Park YM, Lee J, et al. The EMBL-EBI search and sequence analysis tools APIs in 2019. Nucleic Acids Research. 2019 Jul;47(W1):W636-W641. DOI: 10.1093/nar/gkz268.

[6] Waterhouse A.M., Procter J.B., Martin D.M.A., Clamp M., Barton G.J. (2009) Jalview Version 2-a multiple sequence alignment editor and analysis workbench. Bioinformatics 25: 1189-1191. Pubmed: 19151095 DOI: doi:10.1093/bioinformatics/btp033

[7] Lam, T., Shum, M., Zhu, H., Tong, Y., Ni, X., Liao, Y., Wei, W., Cheung, W., Li, W., Li, L., Leung, G., Holmes, E., Hu, Y. and Guan, Y., 2020. Identifying SARS-CoV-2 related coronaviruses in Malayan pangolins. Nature,.

[8] Zhang, Y., 2020. More Evidence Suggests Pangolins May Have Passed Coronavirus From Bats To Humans. [online] ScienceAlert. Available at: https://www.sciencealert.com/more-evidence-suggests-pangolins-may-have-passed-coronavirus-from-bats-to-humans [Accessed 15 May 2020].

[9] Liu, P., Jiang, J., Wan, X., Hua, Y., Li, L., Zhou, J., Wang, X., Hou, F., Chen, J., Zou, J. and Chen, J., 2020. Are pangolins the intermediate host of the 2019 novel coronavirus (SARS-CoV-2)?. PLOS Pathogens, 16(5), p.e1008421.

Contributions

Daihao Xue:

Dealt with imbalanced data (Oversampling for the minority), dealt with dimension reduction, designed the XGBoost model for predicting the origin of COVID-19. Extracting n-gram of the sequences. EDA of the data distribution.

Junda Li:

Literature review for early project formulation and articles about Pangolin and COVID-19. Naive Bayes, KNN and dummy baseline Benchmark; Naive Bayes model tuning. Exploration of different n-gram for features. Prediction of COVID-19 using KNN model.

Zhuolin Mao:

Gathered data, investigated the background and motivations, investigated the data file preprocessing process and the approaches of feature extraction for the machine learning process, performed multiple sequence alignment and implemented phylogenetic tree tools, wrote part of the report, proofread and organized it.