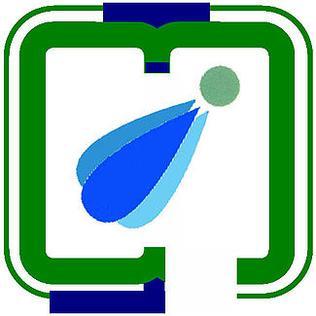
**Indian Institute of Information Technology, Design and Manufacturing, Jabalpur**

****

Spectral Characterization of

SARS-CoV-2 Genome Using

KNN Classifier

BY

YOGESH CHAND UPADHYAY 20BEC126

TARSH 20BCS228

MAYANK SAHU 20BEC063

SAUMYA SRIVASTAVA 20BSM050

NAVNEET KUMAR 20BCS142

# Introduction

COVID-19 is a disease that took the world by storm. It caused more deaths than in world war 2. But they could have been avoided by the application of Biotechnology. If a new disease is detected, we can find its family by applying biotechnology methods. Once a family of a disease is found, we can try out the standard treatment methods of that family. It can also lead to the fastening of the process of vaccine creation. In the case of COVID-19, it belongs to Coronaviridae. Biotechnology methods can be applied in analyzing features of its genome sequence and then feeding these features to a machine learning model. The things that can be predicted are the family of disease, its mortality, and its rate of spreading. In this document, we have discussed various methods by which we can tell whether a disease is under the umbrella of Coronaviridae diseases or not.

# Database

The data has been downloaded from multiple websites. COVID sequences were downloaded from NCBI and INSACOG. Non-COVID sequences like dengue, cancer, etc. (to differentiate from COVID) were downloaded from COSMIC and NCBI. The data files were downloaded in .fasta format. A total of 30 COVID and 25 non-COVID sequences were downloaded.

# Methodology

1. Preprocessing of data was done. The name of the disease was removed from .fasta file as a standard function in MATLAB that only recognizes base pairs (A, G, C, T). Also, extra spaces from the start and end were removed. Then it is imported into MATLAB and converted to a string.
2. Some basic features like the percentage of A, G, C, and T and the percentage of dimers like AC and GT are calculated from MATLAB\_BIOINFORMATICS\_TOOLBOX functions like ntdensity, basecount and dimercount.
3. The first step before doing spectral analysis is to convert our sequence into EIIP scheme.
4. Then the EIIP scheme output is converted to the frequency domain using Discrete Fourier Transform. It is done using fft() inbuilt function of Matlab, which is very fast in implementation.
5. Now our signal is ready for feature extraction. Features like mean, median,peak1, peak2, interquartile ranges, etc. They use inbuilt functions like max, sum, prctile , etc.
6. The features of all the sequences are exported in an excel file.
7. The excel data is converted into a 2-d matrix. The matrix is our data that must be fed into the machine learning model.
8. The data is divided into two parts, one for training and one for testing. The algorithm we use is KNN. In this, results are found using the closest proximity to the trained model's k points.
9. Data is fed into the model and then tested. For the value of k =5, we get an accuracy of 0.9130434782608695.

# Results

For the value of k=5 maximum accuracy is found. The accuracy we get is 0.9130434782608695.

# Conclusion

We can detect whether a disease is a Coronoviridae member or not by doing its base and spectral analysis and optimizing treatment methods, and decreasing fatalities.High cost laboratory methods can be avoided and these methods can produce highly accurate results much faster.

# Codes

The following code is used for preprocessing data and finding the base-related features using the Matlab bioinformatics toolbox. Then it uses a functionsegmentation\_and\_fft which does the spectral characterization of DNA and is explained further.

x = textread('D:\Devsupport\covid\_data\sequence(19).fasta','%s');

ans=[];

a = char(x);

[r2 c2] = size(a);

c = [];

for k = 1:r2

for m = 1:c2

c = [c a(k,m)];

end

end

newStr = c(2:end);

t=deblank(newStr);

p=basecount(t);

A=p.A;

G=p.G;

C=p.C;

T=p.T;

TOTAL=A+G+C+T;

[Dimers, Percent] = dimercount(t);

features = segmentation\_and\_fft(t);

ans=[ans,A\*100/TOTAL,G\*100/TOTAL,C\*100/TOTAL,T\*100/TOTAL,Percent(13)\*100,Percent(10)\*100,features];

disp(ans) ;

x = textread('D:\Devsupport\covid\_data\sequence(19).fasta','%s');

ans=[];

a = char(x);

[r2 c2] = size(a);

c = [];

for k = 1:r2

for m = 1:c2

c = [c a(k,m)];

end

end

newStr = c(2:end);

t=deblank(newStr);

p=basecount(t);

A=p.A;

G=p.G;

C=p.C;

T=p.T;

TOTAL=A+G+C+T;

[Dimers, Percent] = dimercount(t);

features = segmentation\_and\_fft(t);

ans=[ans,A\*100/TOTAL,G\*100/TOTAL,C\*100/TOTAL,T\*100/TOTAL,Percent(13)\*100,Percent(10)\*100,features];

disp(ans) ;

The following code does spectral characterization of the DNA data. It converts the discrete signal into its frequency domain and then finds certain features using inbuilt Matlab functions.

function features = segmentation\_and\_fft(x)

c=char(x);

spectrum = [];

spectrumTwo = [];

eiip = [];

for i=1:length(c)

if (c(i)=='A') || (c(i)=='a')

eiip=[eiip,0.1260];

elseif (c(i)=='C') || (c(i)=='c')

eiip=[eiip,0.1340];

elseif (c(i)=='T') || (c(i)=='t')

eiip=[eiip,0.1335];

else

eiip=[eiip,0.0806];

end

end

Feiip = fft(eiip);

for i=1:length(c)

specTotal = (abs(Feiip(i))^2);

specTwo = (abs(Feiip(i)));

spectrum=[spectrum,specTotal];

spectrumTwo=[spectrumTwo,specTwo];

end

spectrum=10\*log10(spectrum);

features = [];

average = sum((spectrum))/length(spectrum);

features=[features,average];

%medians = median(spectrum,'all');

%features=[features,medians];

maximum = max((spectrum));

features=[features,maximum];

minimum = min((spectrum));

features=[features,minimum];

peak = (length(spectrum)/3)/(average);

features=[features,peak];

%peak\_two = (length(spectrumTwo)/3)/(mean(spectrumTwo,'all'));

%features=[features,peak\_two];

stddev = std((spectrum));

features=[features,stddev];

percentile15 = prctile(spectrum, 15);

features=[features,percentile15];

percentile25 = prctile(spectrum, 25);

features=[features,percentile25];

percentile50 = prctile(spectrum, 50);

features=[features,percentile50];

percentile75 = prctile(spectrum, 75);

features=[features,percentile75];

amplitude = maximum - minimum;

features=[features,amplitude];

variance = var((spectrum));

features=[features,variance];

interquartile\_range =(prctile(spectrum, 75)) - (prctile(spectrum, 25));

features=[features,interquartile\_range];

semi\_interquartile\_range = ((prctile(spectrum, 75)) - (prctile(spectrum, 25)))/2;

features = [features, semi\_interquartile\_range];

coefficient\_of\_variation = stddev/average;

features=[features, coefficient\_of\_variation];

kurtosis = ((prctile(spectrum, 75)) - (prctile(spectrum, 25))) / (2 \* ((prctile(spectrum, 90)) - (prctile(spectrum, 10))));

features=[features, kurtosis];

fre = 1:length(spectrum);

end

After finding the characteristics of DNA data, we are ready to feed it into the model.

The following is the architecture of the KNN Model. X is the matrix of features, and Y is whether the corresponding X index is COVID or not.

import numpy as np

X=[[38.5266,22.4638,14.9758,24.0338,09.2426,0.2567,-5.7609,40.9898,-30.8689,-57.7461,5.9111,-11.6967,-9.1436,-4.855,-2.0228,71.8587,34.9414,7.1208,3.5604,-1.0261,0.2596,],[26.2260,22.1748,22.8145,28.7846,06.4103,05.5556,-09.3115,35.0010,-39.5484,-16.7893,6.0472,-14.9352,-12.5138,-08.4980,-5.4933,74.5494,36.5683,7.0206,3.5103,-0.6494,0.2462,],[39.6319,21.4724,15.5828,23.3129,10.4636,0.5298,-5.7629,40.9794,-32.6531,-57.7251,5.9486,-11.1921,-8.6167,-5.0548,-1.8941,73.6325,35.3864,6.7226,3.3613,-1.0322,0.2551,],[24.5203,26.8657,26.4392,22.1748,04.7009,05.7692,-08.9005,34.8302,-30.3398,-17.5646,6.2833,-14.9332,-12.5933,-07.4909,-4.6634,65.1700,39.4803,7.9299,3.9650,-0.7060,0.2846,],[38.9626,21.2304,15.1990,24.6080,10.2828,0.3856,-5.8539,41.0325,-31.7680,-56.8286,5.9582,-11.3086,-9.3325,-5.456,-1.9074,72.8006,35.5000,7.4251,3.7125,-1.0178,0.2746,],[38.7173,20.4276,16.0333,24.8219,09.3052,0.4963,-6.0850,41.1002,-48.2712,-54.6696,6.3497,-11.4917,-8.7121,-5.5321,-2.3012,89.3714,40.3193,6.4109,3.2055,-1.0435,0.2404,],[27.9318,20.0426,17.4840,34.5416,09.1880,03.4188,-09.3388,35.0713,-31.7945,-16.7403,5.6881,-15.0125,-12.8035,-08.2446,-5.3010,66.8657,32.3547,7.5025,3.7512,-0.6091,0.2943,],[18.4100,30.3347,29.0795,22.1757,5.2411,8.5954,-8.2263,34.8940,-27.6611,-19.3687,5.9375,-13.7362,-11.2512,-7.4748,-4.1153,62.5551,35.2535,7.1359,3.5680,-0.7218,0.2671,],[27.0771,27.9139,22.4148,22.5941,5.6220,2.8110,-3.3553,45.8199,-27.3020,-166.2041,5.9052,-9.1158,-6.3641,-2.4414,0.5628,73.1219,34.8708,6.9268,3.4634,-1.7599,0.2424,],[24.0938,21.3220,30.4904,24.0938,07.2650,06.1966,-09.5727,35.0480,-32.6725,-16.3312,6.3237,-15.7984,-12.9548,-08.5240,-5.2106,67.7205,39.9890,7.7442,3.8721,-0.6606,0.2614,],[37.5610,22.5610,13.6585,26.2195,09.9476,0.6545,-5.4210,40.9670,-30.9652,-61.3659,5.5352,-10.9562,-8.3909,-4.765,-1.8794,71.9322,30.6382,6.5114,3.2557,-1.0211,0.2555,],[39.4737,20.4545,15.7895,24.2823,09.4937,0.2532,-5.9104,41.0776,-28.3687,-56.2850,5.9197,-11.2175,-8.5148,-5.3156,-2.0347,69.4463,35.0426,6.4801,3.2401,-1.0016,0.2473,],[33.4755,20.0426,15.7783,30.7036,10.2564,03.2051,-09.7299,35.0407,-27.6184,-16.0673,5.8011,-15.0012,-12.8354,-09.1136,-5.5181,62.6592,33.6532,7.3173,3.6586,-0.5962,0.2691,],[27.0771,27.9139,22.4746,22.5344,5.6220,2.8708,-3.3549,45.8199,-27.2205,-166.2247,5.9038,-9.1278,-6.3582,-2.4438,0.5667,73.0404,34.8546,6.9249,3.4625,-1.7598,0.2424,],[26.2260,22.1748,22.8145,28.7846,06.4103,05.5556,-09.3115,35.0010,-39.5484,-16.7893,6.0472,-14.9352,-12.5138,-08.4980,-5.4933,74.5494,36.5683,7.0206,3.5103,-0.6494,0.2462,],[26.6525,21.3220,23.2409,28.7846,06.4103,05.1282,-09.3284,35.0315,-31.2794,-16.7588,6.0259,-14.5177,-12.4992,-08.4481,-5.3351,66.3109,36.3116,7.1642,3.5821,-0.6460,0.2623,],[39.0832,20.2654,15.6815,24.9698,09.9872,0.6402,-5.7848,41.0647,-22.6706,-57.5067,5.6719,-11.6183,-8.9837,-5.1594,-1.9428,63.7353,32.1706,7.0409,3.5204,-0.9805,0.2578,],[39.4737,20.4545,15.7895,24.2823,09.4937,0.2532,-5.9104,41.0776,-28.3687,-56.2850,5.9197,-11.2175,-8.5148,-5.3156,-2.0347,69.4463,35.0426,6.4801,3.2401,-1.0016,0.2473,],[12.1535,36.8870,31.9829,18.9765,02.9915,13.2479,-08.3263,34.5031,-33.0635,-18.7758,6.3339,-14.4529,-11.5514,-07.5685,-4.6676,67.5666,40.1186,6.8838,3.4419,-0.7607,0.2333,],[31.6239,09.1880,19.0171,40.1709,12.8480,01.7131,-11.9866,35.4373,-33.1551,-13.0145,5.6863,-17.0775,-15.1680,-11.3311,-8.4968,68.5924,32.3338,6.6712,3.3356,-0.4744,0.2612,],[12.1535,36.8870,31.9829,18.9765,02.9915,13.2479,-08.3263,34.5031,-33.0635,-18.7758,6.3339,-14.4529,-11.5514,-07.5685,-4.6676,67.5666,40.1186,6.8838,3.4419,-0.7607,0.2333,],[19.6396,27.9279,30.8108,21.6216,4.6931,8.8448,-8.1581,36.2798,-66.7596,-22.6769,7.0351,-13.4532,-11.2374,-7.4158,-3.4771,103.0393,49.4933,7.7603,3.8802,-0.8624,0.2570,],[38.1065,21.7751,15.1479,24.9704,10.3960,0.6188,-6.1093,41.0658,-29.1055,-54.4522,6.0349,-11.6764,-9.3375,-5.5167,-2.1141,70.1713,36.4202,7.2234,3.6117,-0.9878,0.2654,],[39.1924,20.6651,15.5582,24.5843,10.1124,0.2497,-5.8587,41.0897,-30.5434,-56.7813,5.6859,-10.8886,-9.0723,-5.3336,-2.3938,71.6331,32.3292,6.6784,3.3392,-0.9705,0.2593,],[38.8688,20.9386,15.5235,24.6691,10.0637,0.2548,-5.9511,41.0490,-30.6676,-55.8998,5.8880,-11.5645,-9.3587,-5.3929,-2.3334,71.7166,34.6685,7.0253,3.5127,-0.9894,0.2500,],[19.8077,29.4231,29.4231,21.3462,5.0096,8.0925,-7.8578,35.6538,-30.2738,-22.0589,5.9752,-13.4623,-10.7164,-6.7377,-3.7609,65.9276,35.7029,6.9556,3.4778,-0.7604,0.2617,],[18.5897,33.1197,29.0598,19.2308,04.2827,10.4925,-08.1624,34.5985,-38.8850,-19.1121,6.2260,-14.1320,-11.6065,-06.7986,-4.2838,73.4835,38.7636,7.3228,3.6614,-0.7628,0.2638,],[20.9251,27.7533,28.4141,22.9075,5.7395,5.7395,-8.9769,34.5338,-33.1701,-16.8581,6.5283,-15.3969,-11.8784,-7.9650,-4.5001,67.7040,42.6181,7.3782,3.6891,-0.7272,0.2520,],[38.6714,20.8778,14.8280,25.6228,10.4348,0.4969,-5.9644,41.0877,-29.5771,-55.7750,5.9631,-11.8140,-9.0660,-5.408,-1.9235,70.6648,35.5582,7.1425,3.5713,-0.9998,0.2511,],[21.5491,26.1503,30.5215,21.7791,4.4513,7.2909,-4.4803,43.7581,-42.6038,-97.0178,6.2476,-10.0788,-7.5932,-3.4405,-0.3111,86.3619,39.0329,7.2821,3.6411,-1.3945,0.2549,],[30.9829,18.3761,17.3077,33.3333,09.4218,02.7837,-09.9452,35.0994,-31.4066,-15.6860,6.1610,-16.5603,-13.0221,-08.9677,-5.9522,66.5061,37.9580,7.0699,3.5349,-0.6195,0.2572,],[38.0835,21.2531,16.8305,23.8329,09.2348,0.7916,-5.6386,40.9914,-28.2703,-58.9979,5.7358,-11.0585,-8.8422,-4.9684,-1.9847,69.2617,32.8996,6.8575,3.4287,-1.0172,0.2578,],[17.3077,25.2137,30.5556,26.9231,05.1392,09.4218,-08.2123,34.9171,-30.4576,-18.9959,5.2211,-12.9713,-10.8513,-07.7450,-4.4737,65.3746,27.2602,6.3776,3.1888,-0.6358,0.2702,],[24.5203,26.8657,26.4392,22.1748,04.7009,05.7692,-08.9005,34.8302,-30.3398,-17.5646,6.2833,-14.9332,-12.5933,-07.4909,-4.6634,65.1700,39.4803,7.9299,3.9650,-0.7060,0.2846,],[16.8443,36.4606,35.1812,11.5139,01.2821,09.4017,-07.9300,34.4946,-27.8045,-19.7141,5.8549,-13.0510,-11.1301,-07.2353,-3.9226,62.2991,34.2798,7.2074,3.6037,-0.7383,0.2598,],[40.4015,19.6989,15.1819,24.7177,12.2392,0.4172,-5.8410,40.9797,-29.4640,-56.9533,5.8921,-11.4123,-8.8396,-5.2583,-2.2502,70.4437,34.7166,6.5894,3.2947,-1.0087,0.2497,],[33.4755,20.0426,15.7783,30.7036,10.2564,03.2051,-09.7299,35.0407,-27.6184,-16.0673,5.8011,-15.0012,-12.8354,-09.1136,-5.5181,62.6592,33.6532,7.3173,3.6586,-0.5962,0.2691,],[38.3055,22.0764,15.8711,23.7470,10.4141,0.7528,-5.8867,41.0340,-33.0265,-56.5113,6.0770,-11.5724,-9.2291,-4.9345,-2.0834,74.0605,36.9295,7.1457,3.5728,-1.0323,0.2565,],[15.3518,33.9019,36.2473,14.4989,02.9915,11.3248,-08.2437,34.6068,-34.5414,-18.9639,6.1527,-13.6739,-11.3832,-07.5444,-4.1035,69.1482,37.8555,7.2797,3.6399,-0.7463,0.2581,],[38.6282,21.0590,15.6438,24.6691,10.8696,0.3836,-5.9570,41.0461,-30.1248,-55.8446,5.9726,-11.8507,-9.2294,-5.3636,-2.0093,71.1709,35.6717,7.2201,3.6100,-1.0026,0.2578,],[15.3518,33.9019,36.2473,14.4989,02.9915,11.3248,-08.2437,34.6068,-34.5414,-18.9639,6.1527,-13.6739,-11.3832,-07.5444,-4.1035,69.1482,37.8555,7.2797,3.6399,-0.7463,0.2581,],[38.3315,21.1950,16.2345,24.2390,09.3785,0.3390,-6.3719,40.7721,-35.3054,-48.5462,5.8594,-11.9310,-9.3498,-5.6189,-2.5801,76.0775,34.3331,6.7696,3.3848,-0.9196,0.2512,],[20.8984,25.7813,29.4922,23.8281,5.6751,7.2407,-8.3597,35.6555,-27.8700,-20.4153,5.9306,-13.7435,-11.6472,-7.4652,-4.5178,63.5255,35.1722,7.1294,3.5647,-0.7094,0.2487,],[16.8443,36.4606,35.1812,11.5139,01.2821,09.4017,-07.9300,34.4946,-27.8045,-19.7141,5.8549,-13.0510,-11.1301,-07.2353,-3.9226,62.2991,34.2798,7.2074,3.6037,-0.7383,0.2598,],[24.0938,21.3220,30.4904,24.0938,07.2650,06.1966,-09.5727,35.0480,-32.6725,-16.3312,6.3237,-15.7984,-12.9548,-08.5240,-5.2106,67.7205,39.9890,7.7442,3.8721,-0.6606,0.2614,],[38.9499,20.6349,15.6288,24.7863,11.0092,0.3932,-5.7419,40.9978,-26.8505,-57.7048,5.7600,-10.8953,-8.9561,-5.0715,-2.1561,67.8482,33.1774,6.8000,3.4000,-1.0032,0.2588,],]

X=np.array(X)

y=[0,1,0,1,0,0,1,1,1,1,0,0,1,1,1,1,0,0,1,1,1,1,0,0,0,1,1,1,0,1,1,0,1,1,1,0,1,0,1,0,1,0,1,1,1,0,]

y=np.array(y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size = 0.5, random\_state=42)

knn = KNeighborsClassifier(n\_neighbors=5)

knn.fit(X\_train, y\_train)

# Calculate the accuracy of the model

print(knn.score(X\_test, y\_test))

The results are as follows.



# References

1. <https://academic.oup.com/bib/article/22/2/1197/5892352>
2. <https://www.ncbi.nlm.nih.gov/>
3. <https://cancer.sanger.ac.uk/cosmic>
4. <https://dbtindia.gov.in/insacog>
5. <http://www.mathworks.in/help/matlab/ref/fft.html>