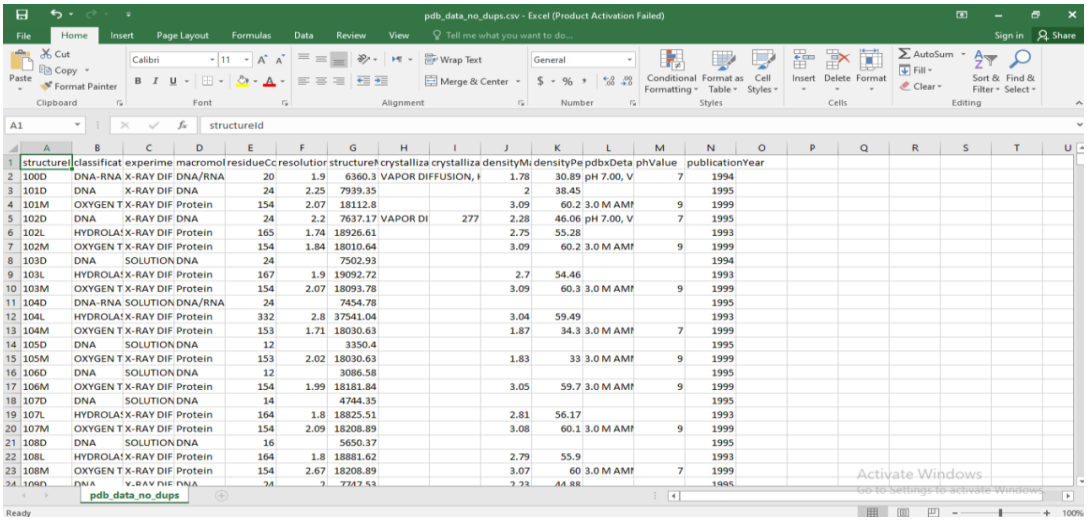
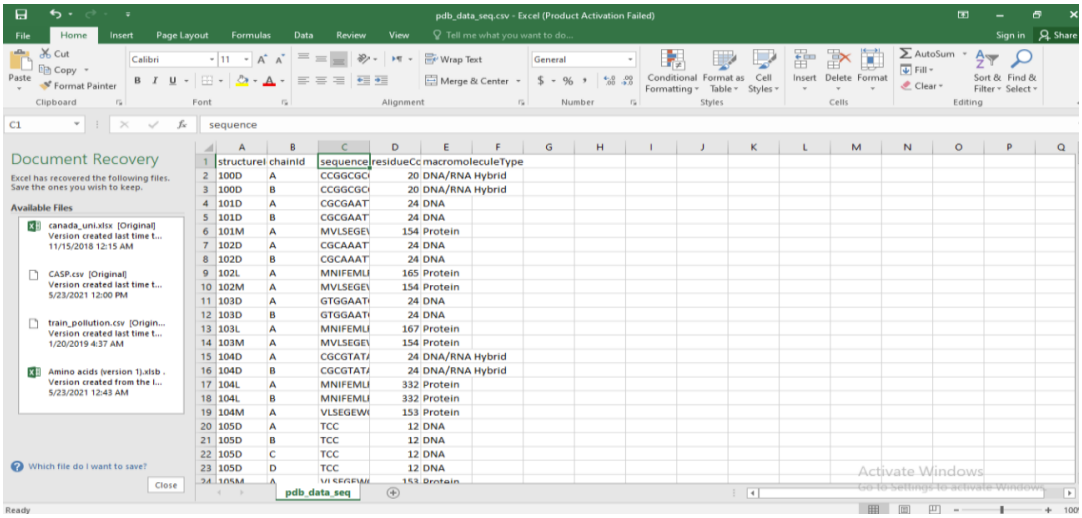
Protein-protein interaction prediction

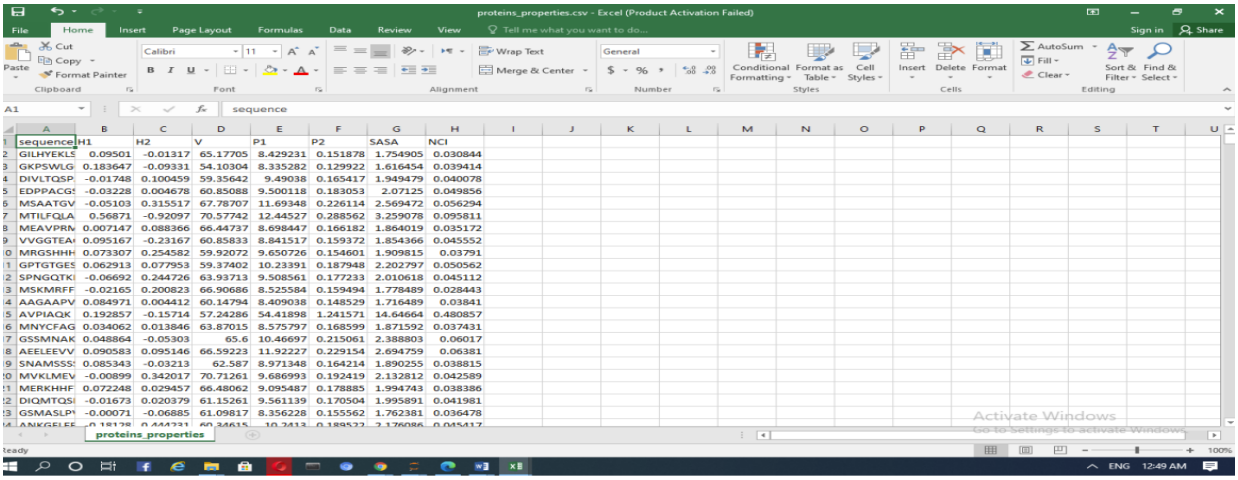
(Research report)

At first, I tried to predict protein-protein interactions using the features of their amino acids. These features are H1, hydrophobicity; H2, hydrophilicity; V, volume of side chains; P1, polarity; P2, polarizability; SASA, solvent accessible surface area; NCI, net charge index of side chains. The protein sequences are gathered from two datasets. The following images show the structure of these datasets.

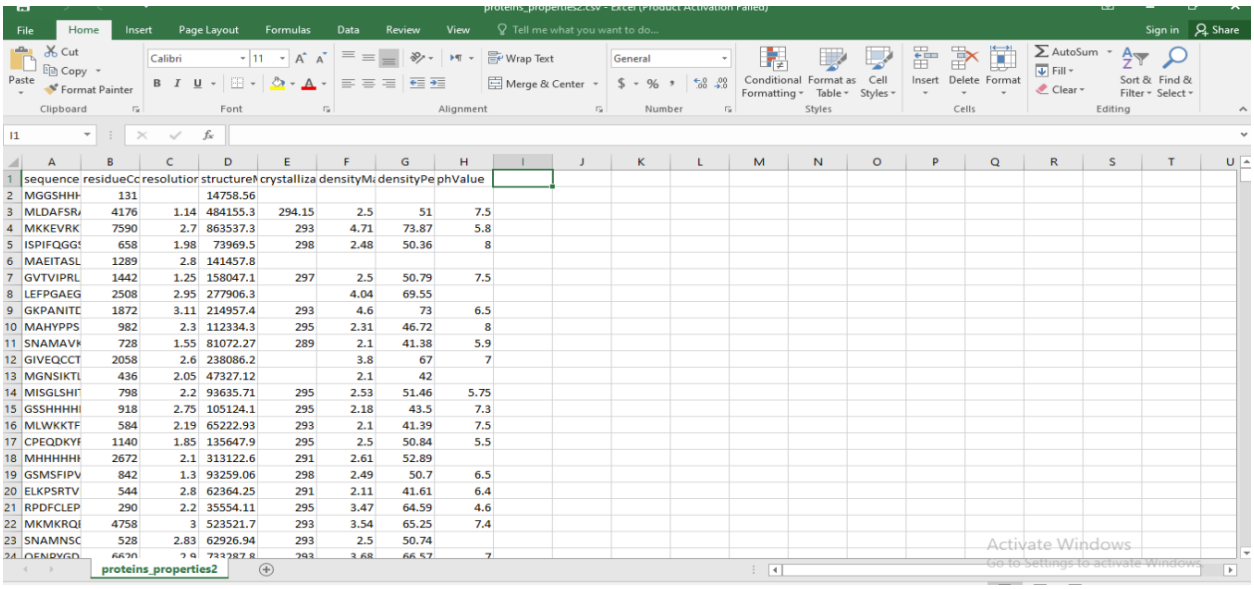




I joined the two datasets since they have the same structureId column. I used 10390 data. I used the sequence column to compute the features for each protein sequence. I averaged the feature value of amino acids for each sequence to compute the feature value for each protein sequence. I made a csv file including the proteins and their features. The following image shows the CSV file.



I used scikit-learn library in python to implement the k-means algorithm and classify the sequences. K-means algorithm starts with some data vectors as centroids of clusters, then assigns each vector to the nearest centroid. The new centroid of the cluster will be the mean of the vectors of the cluster. This process will continue until the centroids of clusters will not be changed or we can repeat the process for a specific number of iterations. I used the silhouette score to measure the performance of k-means algorithm on the dataset. The k-means has the best performance when k=3 but it is not realistic since we have around 900 classes of proteins. When I assumed k=916, the silhouette score was 0.3. I also classified protein sequences based on other features (residueCount, resolution, structureMolecularWeight, crystallizationTempK, densityMatthews, densityPercentSol, phValue) using k-means algorithm. The following image shows the CSV file. I filled the missing values with the average of existing values in each column before performing k-means algorithm.



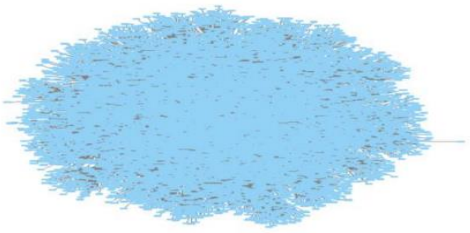
I assumed k=934 since there were 934 types of proteins. The silhouette score was 0.56.

After investigating different references, I concluded that computational methods can be helpful in predicting protein-protein interactions. The computational methods can be divided into three categories. The network-based approach, context aware and specialized methods. The network-based approach methods identify the clusters of proteins that have the same biological function in the network. These methods can be divided into two categories: Divisive methods, Agglomerative methods. The divisive methods divide the network into some subgraphs. The Agglomerative methods start with small subgraphs and then grow these subgraphs to identify cluster.

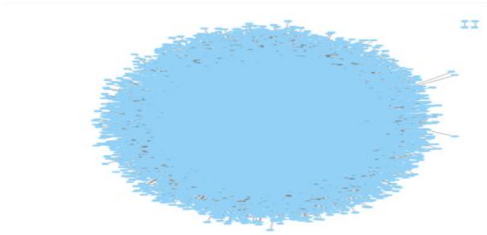
One of the software tools for analyzing and predicting protein-protein interactions is Cytoscape. It has different plug-ins. I imported the network of the database of interacting proteins. I analyzed the network and gained different parameters such as number of nodes, number of edges, average number of neighbors, clustering coefficient, network density and connected components. I downloaded and installed plug-in related to clusterMaker. One of the most famous divisive methods is MCL (Marcov clustering algorithm) and one of the most famous agglomerative methods is MCODE (Molecular complex detection). After performing MCL on the network, 289 clusters were discovered. After performing MCODE, 133 clusters were identified.

According to a paper that Dr. Asgari suggested, I downloaded some datasets for training and evaluation. The datasets include interactions between various protein viruses. The training datasets include 1300 interactions in Hepatitis, 5966 interactions in Herpes, 9880 interactions in HIV, 3044 interactions in Influenza and 5099 interactions in Papilloma. The test datasets include 927 interactions in Dengue, 709 interactions in Zika and 586 interactions of SARS_CoV_2. I have used Cytoscape to draw the protein-protein interaction network graphs of these datasets.

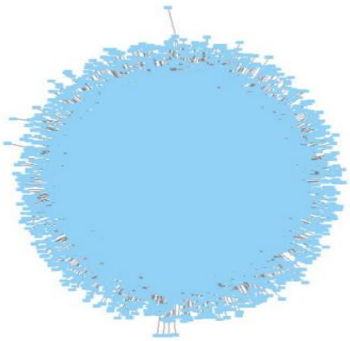
Hepatitis virus:



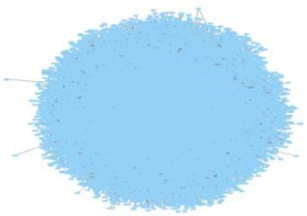
Herpes virus:



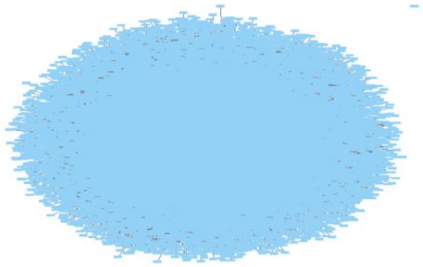
HIV virus:



Influenza virus:



Papilloma virus:



After analyzing the networks, the following statistics were revealed.

Analyzer

Hepatitis_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	10287
Number of edges	14299
Avg. number of neighbors	2.780
Network diameter	7
Network radius	5
Characteristic path length	4.306
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	5.256
Network centralization	0.032
Connected components	1
Analysis time (sec)	8.654

Analyzer

Herpes_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	19845
Number of edges	65625
Avg. number of neighbors	6.615
Network diameter	7
Network radius	5
Characteristic path length	3.958
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	3.297
Network centralization	0.019
Connected components	3
Analysis time (sec)	60.328

Analyzer

HIV_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	20464
Number of edges	108679
Avg. number of neighbors	10.621
Network diameter	6
Network radius	4
Characteristic path length	3.796
Clustering coefficient	0.000
Network density	0.001
Network heterogeneity	4.172
Network centralization	0.041
Connected components	1
Analysis time (sec)	85.471

Analyzer

Influenza_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	16377
Number of edges	33483
Avg. number of neighbors	4.089
Network diameter	8
Network radius	5
Characteristic path length	3.927
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	5.826
Network centralization	0.030
Connected components	1
Analysis time (sec)	28.782

Papilloma_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	18848
Number of edges	52733
Avg. number of neighbors	5.596
Network diameter	6
Network radius	4
Characteristic path length	3.843
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	6.711
Network centralization	0.044
Connected components	2
Analysis time (sec)	46.013

Network diameter is the longest of all the shortest paths in the network.

The following equation defines the clustering coefficient of a node.

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

Where e_i shows the number of existing interactions between the neighbors of the node and k_i shows the number of neighbors of the node.

Clustering coefficient of a network is the average of clustering coefficient of its nodes.

Network radius is the number of nodes in a shortest path of the network.

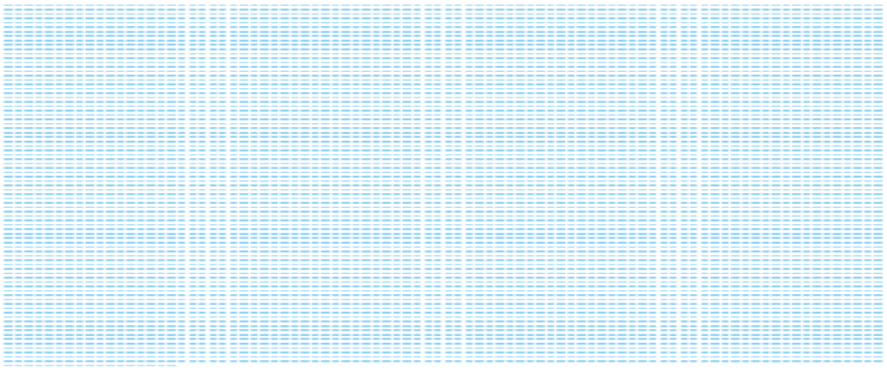
Closeness centrality for a node x is defined as the following equation:

$$C(x) = \frac{N-1}{\sum_y d(y,x)}$$

Where N is the number of nodes and d (y, x) is the distance between vertices y and x.

Heterogeneity of networks refers to networks including entities of multiple types and their relations.

I got the intersection of the networks in Cytoscape to find the overlaps of the networks of these five datasets. The following image shows the intersection of these 5 networks:



There are 7069 nodes (proteins) in the intersection of the networks but there isn't any edge (a pair of proteins that interact with each other).

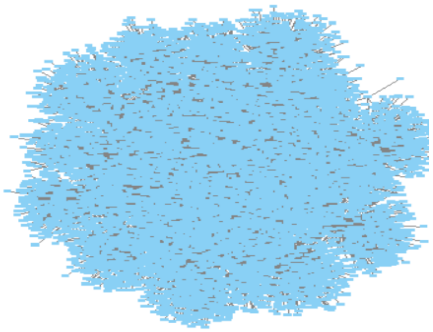
To assess the quality of the networks, I used clusterMaker plug-in (community cluster (GLay)) to cluster the network and found the modularity of the networks. The modularity of a network refers to the strength of a network to be divided into modules (clusters). The following table shows the number of clusters and the modularity for each network.

	Number of clusters	Modularity
Hepatitis	65	0.723
Herpes	38	0.389
HIV	24	0.337
Influenza	48	0.529
Papilloma	37	0.416

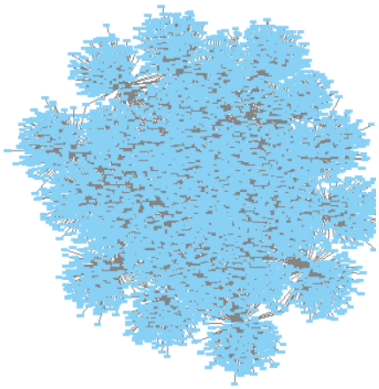
We can see the modularity of the network of the Hepatitis virus is the highest.

The following images also show the networks of the test datasets.

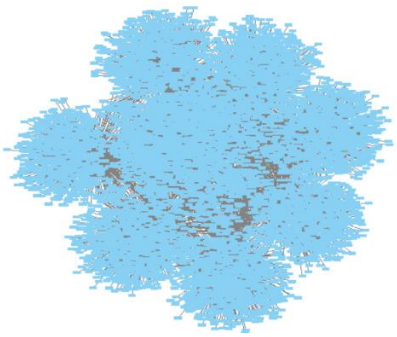
Dengue:



SARS-CoV-2:



Zika:



I also analyzed the networks of the test datasets and the following statistics were revealed.

Analyzer

DENV_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	8027
Number of edges	10196
Avg. number of neighbors	2.540
Network diameter	6
Network radius	3
Characteristic path length	3.839
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	9.851
Network centralization	0.090
Connected components	1
Analysis time (sec)	4.421

Analyzer

SARS2_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	5359
Number of edges	6247
Avg. number of neighbors	2.331
Network diameter	6
Network radius	3
Characteristic path length	3.901
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	7.444
Network centralization	0.057
Connected components	1
Analysis time (sec)	1.939

Analyzer

ZIKV_protein_pair_label.txt (undirected)

Summary Statistics

Number of nodes	6480
Number of edges	7798
Avg. number of neighbors	2.407
Network diameter	4
Network radius	3
Characteristic path length	3.638
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	14.245
Network centralization	0.164
Connected components	1
Analysis time (sec)	2.515

The following image shows the intersection of the test datasets.

There are 704 nodes (proteins) in the intersection of the test datasets and there isn't any edge (a pair of proteins that interact with each other.)

I also used the clusterMaker plug-in again for test datasets to get their modularity of their networks.

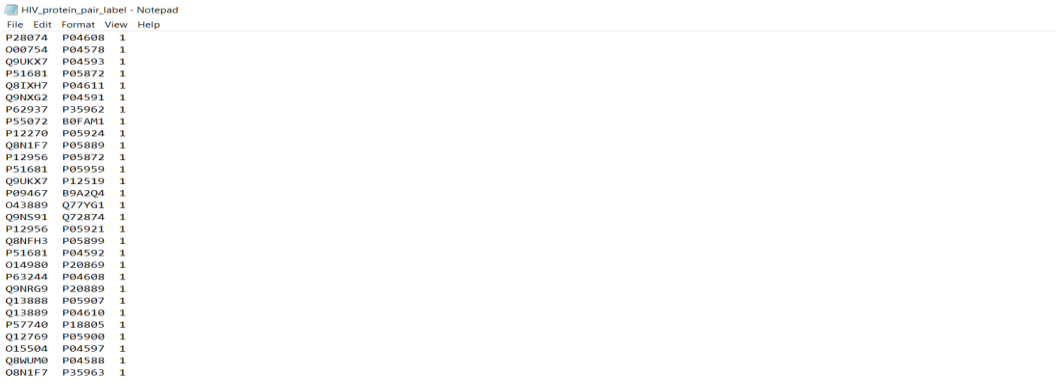
	Number of clusters	Modularity
Dengue	21	0.77
SARS-CoV-2	24	0.833
Zika	8	0.775

We can see that the network of SARS-CoV-2 dataset has the maximum modularity.

The following table also shows the summary of other statistics for each of the datasets.

Dataset	Number of proteins	Numbers of pairs of proteins	Positive pairs	Negative pairs
Hepatitis	10287	14300	1300	13000
Herpes	19845	65626	5966	59660
HIV	20464	108680	9880	98800
Influenza	16377	33480	3044	30440
Dengue	8027	10197	927	9270
Papilloma	18848	52734	5099	50990
SARS-CoV-2	5359	6248	568	5680
Zika	6480	7799	709	7090

After exploring different references and related paper, I concluded that convolutional neural network can work as a model for first predictions. As a first evaluation, I selected the HIV_protein_pair_label dataset to test the convolutional neural network. The first and second columns show the pair of protein sequences. The third column shows whether the pair of proteins can interact together. The following image shows the format of the dataset.


HIV_protein_pair_label - Notepad
File Edit Format View Help
P28074 P04608 1
O00754 P04578 1
Q9UKX7 P04593 1
P51681 P05872 1
Q8IXH7 P04611 1
Q9NXG2 P04591 1
P62937 P35962 1
P55072 B0FAM1 1
P12270 P05924 1
Q8N1F7 P05889 1
P12956 P05872 1
P51681 P05959 1
Q9UKX7 P12519 1
P09467 B9A2Q4 1
O43889 Q77YG1 1
Q9NS91 Q72874 1
P12956 P05921 1
Q8NFH3 P05899 1
P51681 P04592 1
O14980 P20869 1
P63244 P04608 1
Q9NRG9 P20889 1
Q13888 P05907 1
Q13889 P04610 1
P57740 P18805 1
Q12769 P05900 1
O15504 P04597 1
Q8WUM0 P04588 1
O8N1F7 P35963 1

The dataset has 108680 protein pairs and I used 65208 protein pairs as train data frame and 43472 protein pairs as a test data frame. I assigned integers to amino acids in protein sequences and concatenate the pairs of proteins as input. So, for each pair, there is an array including 12 elements. I used the third column as a target variable. Then I made a simple convolutional neural network using

tensorflow.keras library in python and used the model to evaluate the test data frame. The accuracy of the evaluation for the test dataset was 0.91 and it seems that the convolutional neural network can work well for training protein pairs and predicting protein-protein interactions.