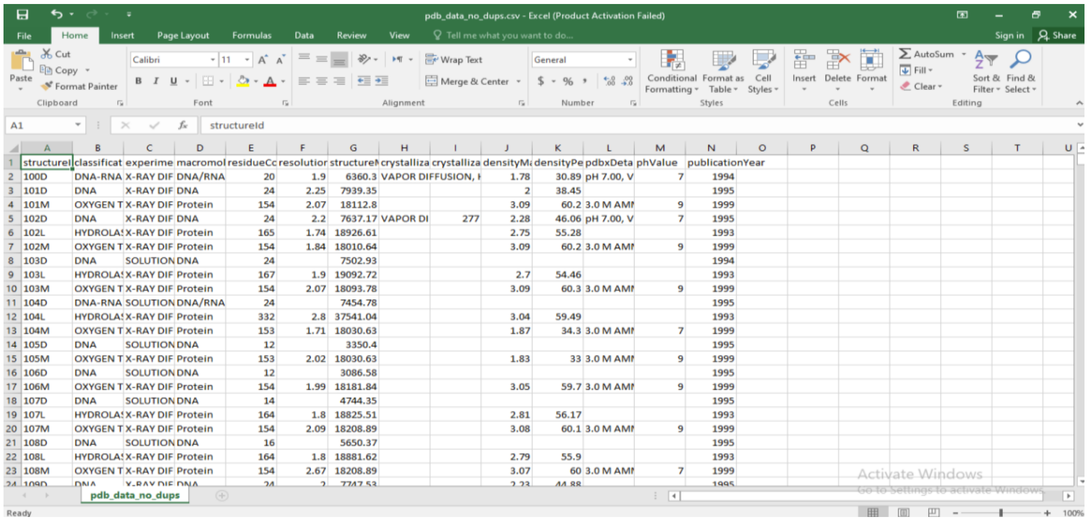
Protein-protein interaction prediction

(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.

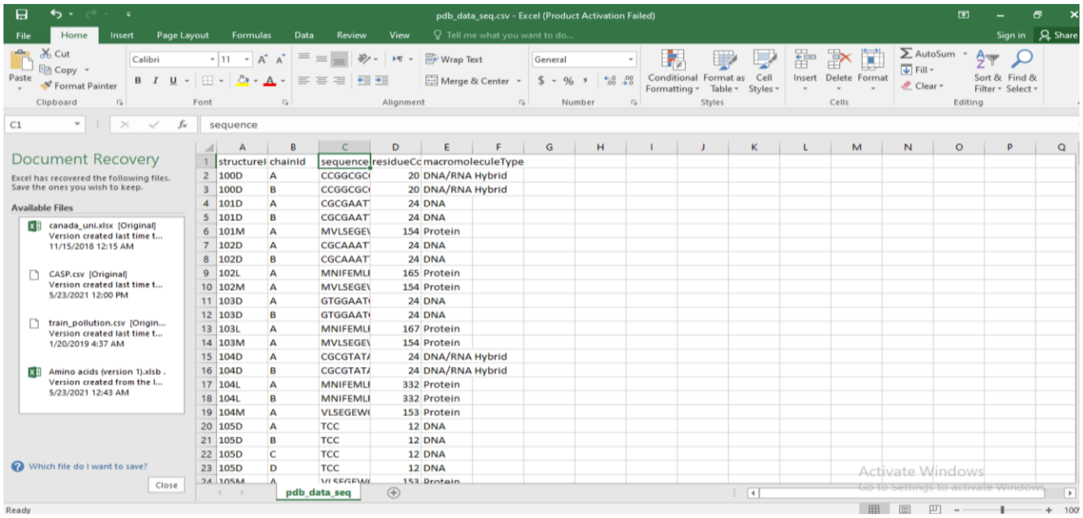


pdb_data_no_dups.csv - Excel (Product Activation Failed)

A1 structureId

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	structureI	classificat	experime	macromol	residueCc	resolutior	structureN	crystalliza	crystalliza	densityMa	densityPe	pdbxDeta	phValue	publicationYear
2	100D	DNA-RNA	X-RAY DIF	DNA/RNA	20	1.9	6360.3	VAPOR DIFFUSION, H		1.78	30.89	pH 7.00, V	7	1994
3	101D	DNA	X-RAY DIF	DNA	24	2.25	7939.35			2	38.45			1995
4	101M	OXYGEN T	X-RAY DIF	Protein	154	2.07	18112.8			3.09	60.2	3.0 M AMN	9	1999
5	102D	DNA	X-RAY DIF	DNA	24	2.2	7637.17	VAPOR DI	277	2.28	46.06	pH 7.00, V	7	1995
6	102L	HYDROLAS	X-RAY DIF	Protein	165	1.74	18926.61			2.75	55.28			1993
7	102M	OXYGEN T	X-RAY DIF	Protein	154	1.84	18010.64			3.09	60.2	3.0 M AMN	9	1999
8	103D	DNA	SOLUTION	DNA	24		7502.93							1994
9	103L	HYDROLAS	X-RAY DIF	Protein	167	1.9	19092.72			2.7	54.46			1993
10	103M	OXYGEN T	X-RAY DIF	Protein	154	2.07	18093.78			3.09	60.3	3.0 M AMN	9	1999
11	104D	DNA-RNA	SOLUTION	DNA/RNA	24		7454.78							1995
12	104L	HYDROLAS	X-RAY DIF	Protein	332	2.8	37541.04			3.04	59.49			1993
13	104M	OXYGEN T	X-RAY DIF	Protein	153	1.71	18030.63			1.87	34.3	3.0 M AMN	7	1999
14	105D	DNA	SOLUTION	DNA	12		3350.4							1995
15	105M	OXYGEN T	X-RAY DIF	Protein	153	2.02	18030.63			1.83	33	3.0 M AMN	9	1999
16	106D	DNA	SOLUTION	DNA	12		3086.58							1995
17	106M	OXYGEN T	X-RAY DIF	Protein	154	1.99	18181.84			3.05	59.7	3.0 M AMN	9	1999
18	107D	DNA	SOLUTION	DNA	14		4744.35							1995
19	107L	HYDROLAS	X-RAY DIF	Protein	164	1.8	18825.51			2.81	56.17			1993
20	107M	OXYGEN T	X-RAY DIF	Protein	154	2.09	18208.89			3.08	60.1	3.0 M AMN	9	1999
21	108D	DNA	SOLUTION	DNA	16		5650.37							1995
22	108L	HYDROLAS	X-RAY DIF	Protein	164	1.8	18881.62			2.79	55.9			1993
23	108M	OXYGEN T	X-RAY DIF	Protein	154	2.67	18208.89			3.07	60	3.0 M AMN	7	1999

pdb_data_no_dups



pdb_data_seq.csv - Excel (Product Activation Failed)

C1 sequence

Document Recovery

Excel has recovered the following files. Save the ones you wish to keep.

Available Files

canada_uni.xlsx [Original]
Version created last time t...
11/15/2018 12:15 AM

CASP.csv [Original]
Version created last time t...
5/23/2021 12:00 PM

train_pollution.csv [Origin...
Version created last time t...
1/20/2019 4:37 AM

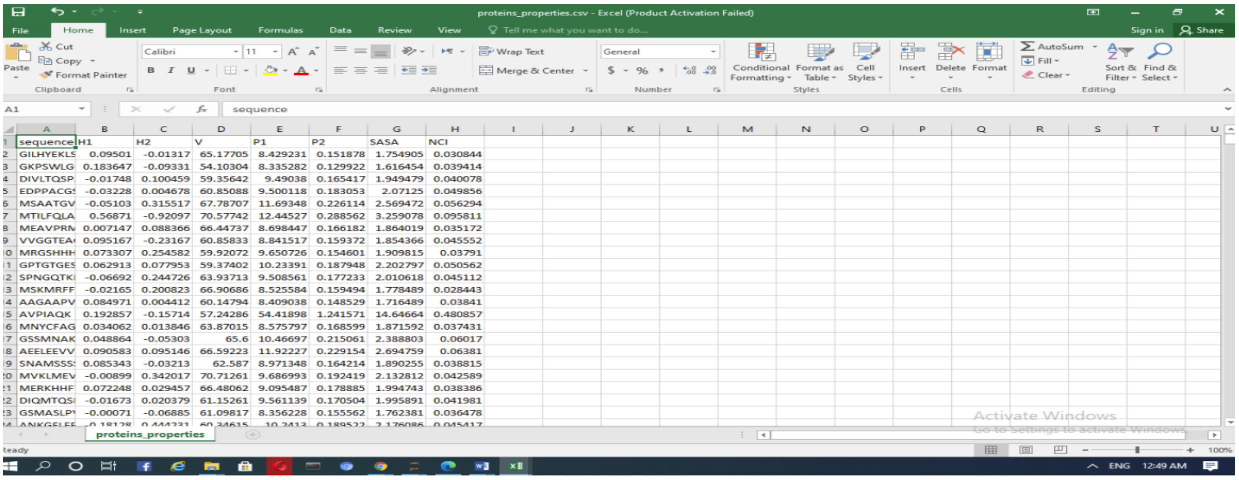
Amino acids (version 1).xlsb .
Version created from the l...
5/23/2021 12:43 AM

Which file do I want to save?

	A	B	C	D	E
1	structureI	chainId	sequence	residueCc	macromoleculeType
2	100D	A	CCGGCGC	20	DNA/RNA Hybrid
3	100D	B	CCGGCGC	20	DNA/RNA Hybrid
4	101D	A	CGCGAAT	24	DNA
5	101D	B	CGCGAAT	24	DNA
6	101M	A	MVLSEGEV	154	Protein
7	102D	A	CGCAAAT	24	DNA
8	102D	B	CGCAAAT	24	DNA
9	102L	A	MNIFEMLI	165	Protein
10	102M	A	MVLSEGEV	154	Protein
11	103D	A	GTGGAAT	24	DNA
12	103D	B	GTGGAAT	24	DNA
13	103L	A	MNIFEMLI	167	Protein
14	103M	A	MVLSEGEV	154	Protein
15	104D	A	CGCGTATA	24	DNA/RNA Hybrid
16	104D	B	CGCGTATA	24	DNA/RNA Hybrid
17	104L	A	MNIFEMLI	332	Protein
18	104L	B	MNIFEMLI	332	Protein
19	104M	A	VLSEGEW	153	Protein
20	105D	A	TCC	12	DNA
21	105D	B	TCC	12	DNA
22	105D	C	TCC	12	DNA
23	105D	D	TCC	12	DNA

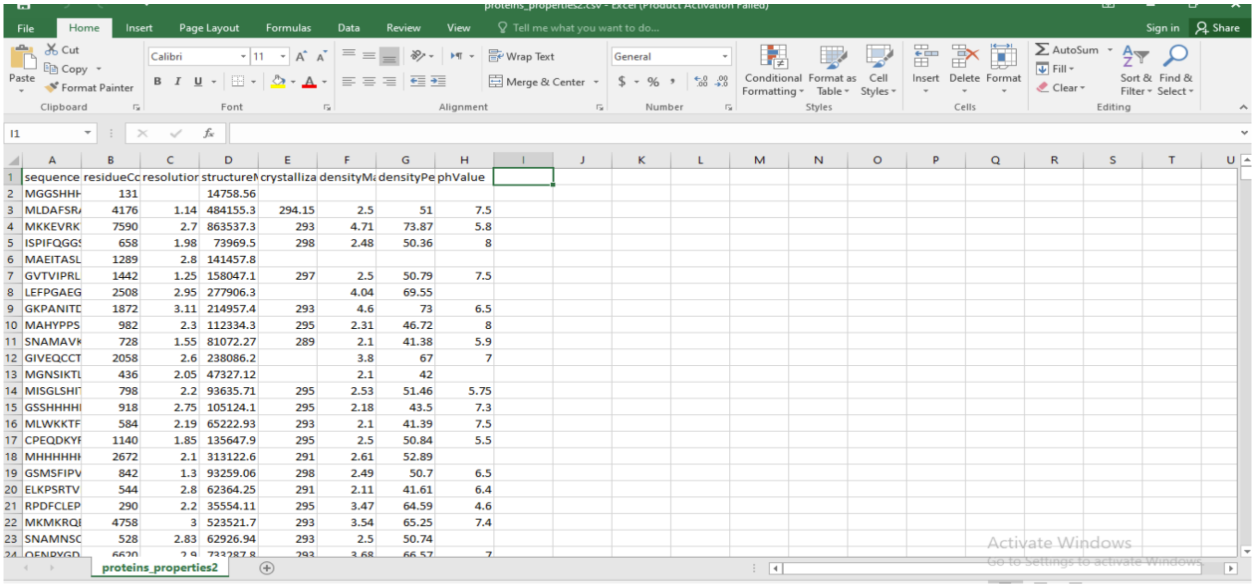
pdb_data_seq

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.



A	B	C	D	E	F	G	H
sequence	H1	H2	V	P1	P2	SASA	NCI
GILHYEKLS	0.09501	-0.01317	65.17705	8.429231	0.151878	1.754905	0.030844
GKPSWLG	0.183647	-0.09331	54.10304	8.335282	0.129922	1.616454	0.039414
DIVLTQSP.	-0.01748	0.100459	59.35642	9.49038	0.165417	1.949479	0.040078
EDPPACG!	-0.03228	0.004678	60.85088	9.500118	0.183053	2.07125	0.049856
MSAATGV	-0.05103	0.315517	67.78707	11.69348	0.226114	2.569472	0.056294
MTILFQLA	0.56871	-0.92097	70.57742	12.44527	0.288562	3.259078	0.095811
MEAVPRN	0.007147	0.088366	66.44737	8.698447	0.166182	1.864019	0.035172
VVGGTEA	0.095167	-0.23167	60.85833	8.841517	0.159372	1.854366	0.045552
MRGSHHH	0.073307	0.254582	59.92072	9.650726	0.154601	1.909815	0.03791
GPTGTGES	0.062913	0.077953	59.37402	10.23391	0.187948	2.202797	0.050562
SPNGQTKI	-0.06692	0.244726	63.93713	9.508561	0.177233	2.010618	0.045112
MSKMRFF	-0.02165	0.200823	66.90686	8.525584	0.159494	1.778489	0.028443
AAGAAPV	0.084971	0.004412	60.14794	8.409038	0.148529	1.716489	0.03841
AVPIAQK	0.192857	-0.15714	57.24286	54.41898	1.241571	14.64664	0.480857
MNYCFAG	0.034062	0.013846	63.87015	8.575797	0.168599	1.871592	0.037431
GSSMNAK	0.048864	-0.05303	65.6	10.46697	0.215061	2.388803	0.06017
AEELEEVV	0.090583	0.095146	66.59223	11.92227	0.229154	2.694759	0.06381
SNAMSSS!	0.085343	-0.03213	62.587	8.971348	0.164214	1.890255	0.038815
MVKLMEV	-0.00899	0.342017	70.71261	9.686993	0.192419	2.132812	0.042589
MERKHHF	0.072248	0.029457	66.48062	9.095487	0.178885	1.994743	0.038386
DIQMTQSI	-0.01673	0.020379	61.15261	9.561139	0.170504	1.995891	0.041981
GSMASLPY	-0.00071	-0.06885	61.09817	8.356228	0.155562	1.762381	0.036478
ANKGELEE	-0.18128	0.444231	60.34615	10.2413	0.189522	2.176086	0.045417

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.


proteins_properties2.csv - Excel (Product Activation Failed)
I1
proteins_properties2

A	B	C	D	E	F	G	H
sequence	residueCc	resolutior	structureN	crystalliza	densityMa	densityPe	phValue
MGGSHHH	131		14758.56				
MLDAFSRA	4176	1.14	484155.3	294.15	2.5	51	7.5
MKKEVRK	7590	2.7	863537.3	293	4.71	73.87	5.8
ISPIFQGG!	658	1.98	73969.5	298	2.48	50.36	8
MAEITASL	1289	2.8	141457.8				
GVTVIPRL	1442	1.25	158047.1	297	2.5	50.79	7.5
LEFPGAEG	2508	2.95	277906.3		4.04	69.55	
GKPANITD	1872	3.11	214957.4	293	4.6	73	6.5
MAHYPPS	982	2.3	112334.3	295	2.31	46.72	8
SNAMAVK	728	1.55	81072.27	289	2.1	41.38	5.9
GIVEQCCT	2058	2.6	238086.2		3.8	67	7
MGNSIKTL	436	2.05	47327.12		2.1	42	
MISGLSHIT	798	2.2	93635.71	295	2.53	51.46	5.75
GSSHHHHI	918	2.75	105124.1	295	2.18	43.5	7.3
MLWKKTF	584	2.19	65222.93	293	2.1	41.39	7.5
CPEQDKYF	1140	1.85	135647.9	295	2.5	50.84	5.5
MHHHHHH	2672	2.1	313122.6	291	2.61	52.89	
GSMSFIPV	842	1.3	93259.06	298	2.49	50.7	6.5
ELKPSRTV	544	2.8	62364.25	291	2.11	41.61	6.4
RPDFCLEP	290	2.2	35554.11	295	3.47	64.59	4.6
MKMKRQI	4758	3	523521.7	293	3.54	65.25	7.4
SNAMNSC	528	2.83	62926.94	293	2.5	50.74	
QENPYGD	6620	2.9	733287.8	293	3.68	66.57	7

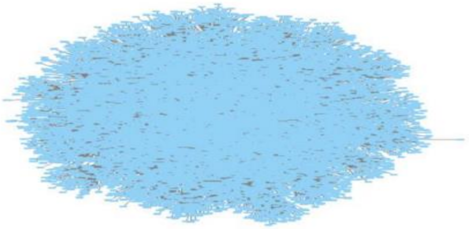
I assumed k=934 since there were 934 types of proteins. The silhouette score was 0.56.

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.

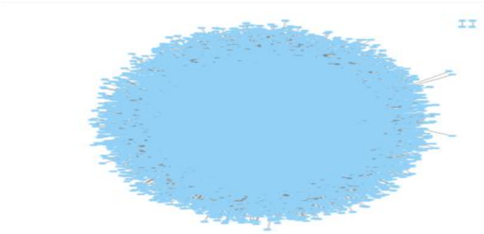
One of the software tools for analyzing and predicting protein-protein interactions is Cytoscape. It has different plug-ins. I imported the network of the database of interacting proteins. I analyzed the network and gained different parameters such as number of nodes, number of edges, average number 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.

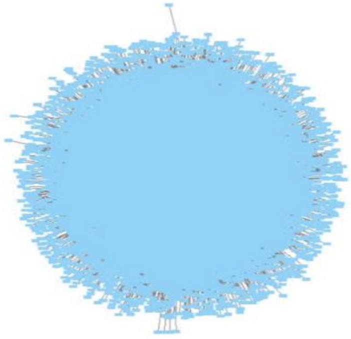
Hepatitis virus:



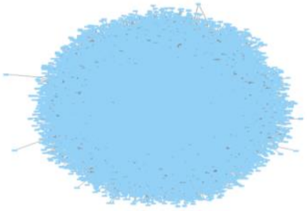
Herpes virus:



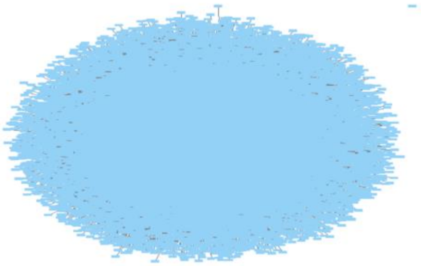
HIV virus:



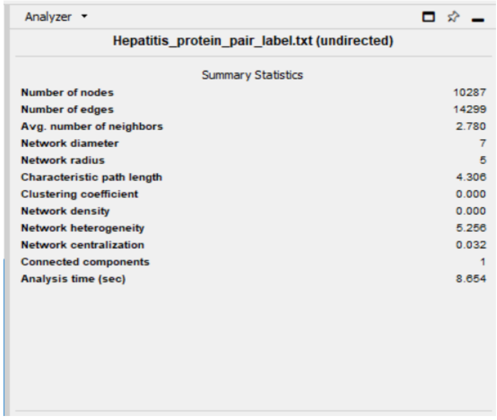
Influenza virus:



Papilloma virus:



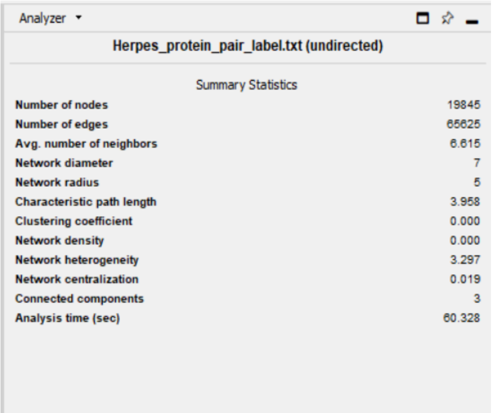
After analyzing the networks, the following statistics were revealed.



Hepatitis_protein_pair_label.txt (undirected)

Summary Statistics

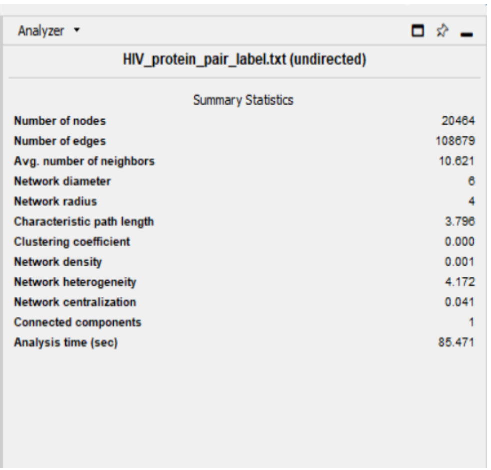
Number of nodes	10287
Number of edges	14299
Avg. number of neighbors	2.780
Network diameter	7
Network radius	5
Characteristic path length	4.306
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	5.256
Network centralization	0.032
Connected components	1
Analysis time (sec)	8.654



Herpes_protein_pair_label.txt (undirected)

Summary Statistics

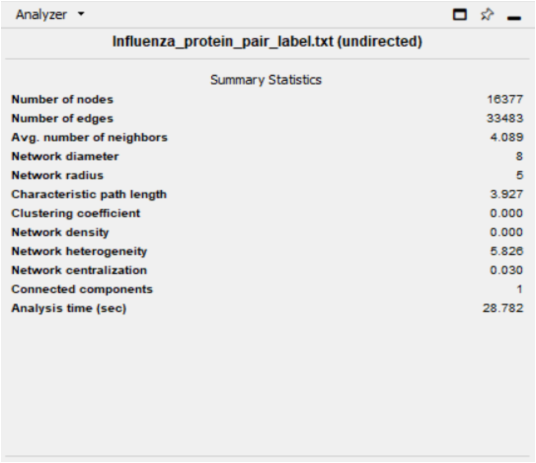
Number of nodes	19845
Number of edges	65625
Avg. number of neighbors	6.615
Network diameter	7
Network radius	5
Characteristic path length	3.958
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	3.297
Network centralization	0.019
Connected components	3
Analysis time (sec)	60.328



HIV_protein_pair_label.txt (undirected)

Summary Statistics

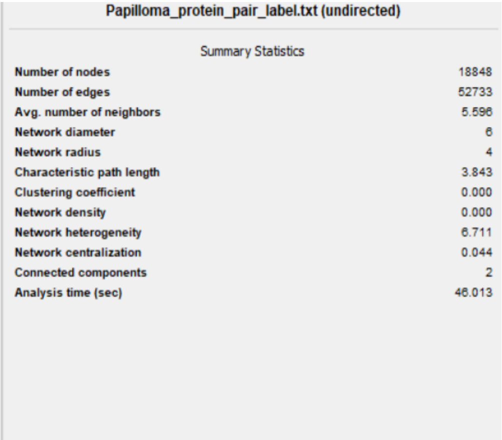
Number of nodes	20464
Number of edges	108679
Avg. number of neighbors	10.621
Network diameter	6
Network radius	4
Characteristic path length	3.796
Clustering coefficient	0.000
Network density	0.001
Network heterogeneity	4.172
Network centralization	0.041
Connected components	1
Analysis time (sec)	85.471



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.

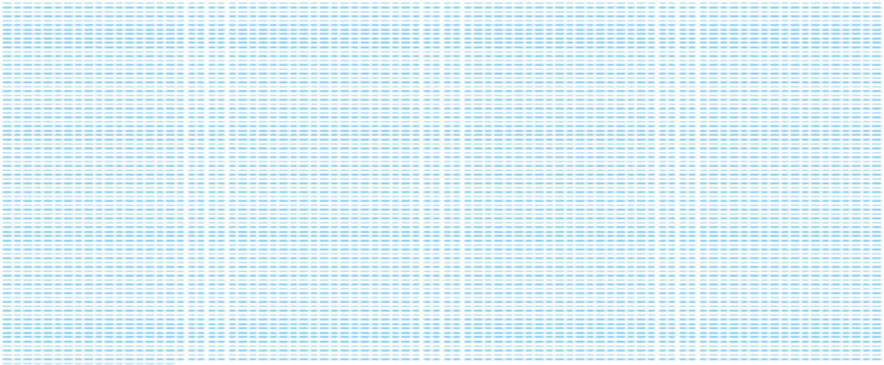
Closeness centrality for a node x is defined as the following equation:

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

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).

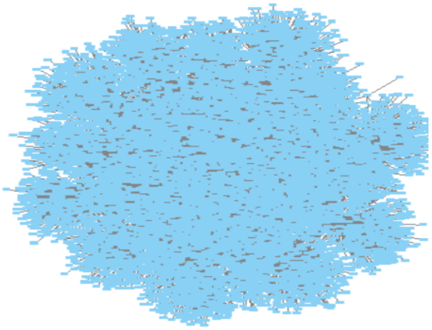
To assess the quality of the networks, I used clusterMaker plug-in (community cluster (GLay)) to cluster the network and found the modularity of the networks. The modularity of a network refers to the strength of a network to be divided into modules (clusters). The following table shows the number of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

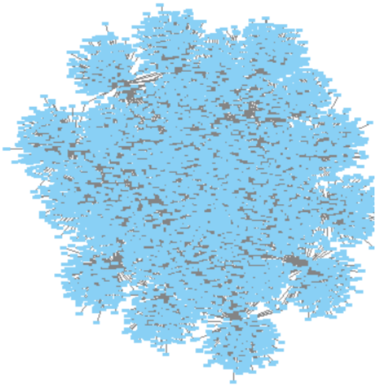
We can see the modularity of the network of the Hepatitis virus is the highest.

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

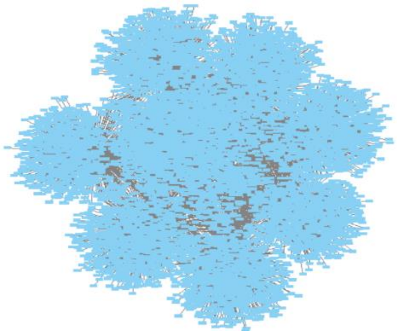
Dengue:



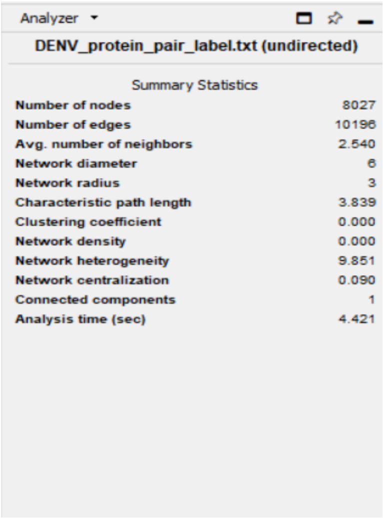
SARS-CoV-2:



Zika:



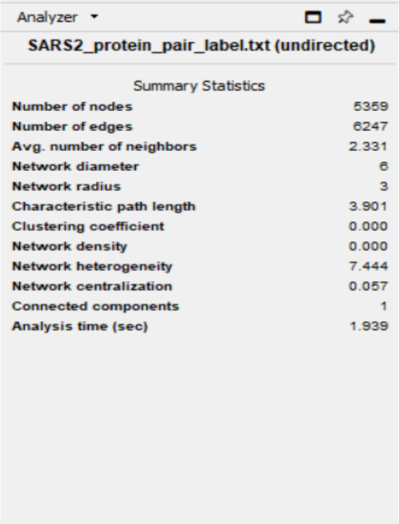
I also analyzed the networks of the test datasets and the following statistics were revealed.



DENV_protein_pair_label.txt (undirected)

Summary Statistics

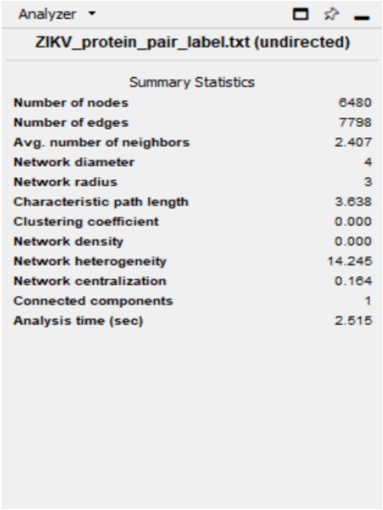
Number of nodes	8027
Number of edges	10196
Avg. number of neighbors	2.540
Network diameter	6
Network radius	3
Characteristic path length	3.839
Clustering coefficient	0.000
Network density	0.000
Network heterogeneity	9.851
Network centralization	0.090
Connected components	1
Analysis time (sec)	4.421



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



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

We have 8 datasets of protein-protein interactions of viruses, but I currently split one of the datasets into train and test datasets (SARS2 protein-protein interaction dataset). At first I removed the redundant pairs. If we have protein-protein interaction A-B and C-D, if (A, C) and (B, D) each has the same cluster (or (A, D) and (B, C) each has the same cluster), they will be redundant and I removed one of them (I used the CD-HIT program to cluster protein sequences). I divided the dataset into 5 folds (4 folds for train set and 1 fold for test dataset). I separated the positive pairs and negative pairs of the train and test datasets. For each positive pair in train set, I selected 10 negative pairs of train set. I also did it for test set. After generating the train and test set, there will be 4775 pairs for train set and 1079 pairs for test set.

I used TF-IDF Vectorizer to represent similarity vectors for SARS2 protein sequences. The following pseudocode shows this.

```
tfidf_vect_ngram_chars =TfidfVectorizer(analyzer='char', ngram_range=(3,3))
x1=tfidf_vect_ngram_chars.fit_transform(texts_AllSeq)
x=np.dot(x1,x1.T)
```

The following matrix shows the TF-IDF vectors of protein sequences. The length of vector of each protein sequence is 5360.

```
[[1.        0.71339452   0.88371699 ... 0.65259757 0.85797354 0.82115687]
```

```
 [0.71339452 1.         0.73815025 ... 0.58022968 0.7305456   0.72052198]
 [0.88371699 0.73815025 1.         ... 0.72283042 0.912652    0.86799822]
 ...
 [0.65259757 0.58022968 0.72283042 ... 1.         0.7334621   0.68576013]
 [0.85797354 0.7305456   0.912652    ... 0.7334621  1.    0.8754899 ]
 [0.82115687 0.72052198 0.86799822 ... 0.68576013  0.8754899  1.        ]]
```

I assigned TF-IDF vectors to the protein sequences in train and test datasets. For each pair, I concatenated two TF-IDF vectors of protein sequences of the pair. So, the length of each element of input is 10720.

I created a basic neural network as a baseline to predict the interactions of the test dataset. It includes the following layers:

one layer: 160 nodes, activation function=relu
one layer: 160 nodes, activation function=relu
one layer: 1 node, activation function=sigmoid

I also set the loss function to binary_crossentropy.

I made prediction with generated train and test datasets and random splitting. It seems that the prediction with generated train and test datasets works a little better. When I changed some parameters (analyze and ngram_range) of TfidfVectorizer like this:

```
tfidf_vect_ngram_chars = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', ngram_range=(3,4))
```

The results varied a little and the value of some of the positive interactions were not near zero.