Solving the problem of antibody grouping based on cross-inhibition index using hierarchical clustering methods

Oleksandr Zelinskyi[a], Vitaliy Horlatch[a], Yuri Lebedin[b] and Yaryna Paslavska[a]

[a] *Ivan Franko National University of Lviv, 1 Universytetska St., Lviv, 79000, Ukraine*
[b] *Xema OY, Myllymäenkatu 21, Lappeenranta, 53550, Finland*

Abstract
Due to the increasing number of viral diseases (including Covid-19), rapid research on possible detection, prevention, and treatment is crucial. Therefore, in this article, the problem of finding two optimal antibodies to any virus (for example, SARS-CoV-2) is considered. In addition, the possible ways of solving this problem using different hierarchical clustering algorithms are described.

Keywords 1
Hierarchical clustering, SARS-CoV-2, antibodies, viruses

1. Introduction

The Covid-19 epidemic has shown that it is still quite difficult for humanity to control and fight acute respiratory viral infections. According to WHO, almost 613 million people worldwide have been infected with COVID-19 and more than 6.5 million people have died due to the disease [3]. However, it is commonly known that this was not the first and probably not the last such pandemic.

Therefore, it is crucial to conduct research as quickly as possible, so that the diseases could be easily detected and treated. The next step is the development of vaccines, as well as tests that show the number of antibodies to a particular virus. It is clear that rapid detection of the disease helps to isolate the spread of the virus and treat a patient more effectively, and vaccination improves immunity to a particular virus and reduces the likelihood of negative (including fatal) consequences.

Nowadays, computers are a very powerful tool that allows solving not only mathematical problems, but also biological, chemical, and medical. Different types of models and algorithms (including machine learning algorithms) are used for that purpose. Moreover, the usage of computers helps scientists to reduce the number of experiments and routine work in laboratories around the world.

The purpose of this work is to consider the problem of finding two optimal antibodies to any virus (for example, the SARS-CoV-2) and propose possible ways to solve it using machine learning algorithms, more precisely agglomerative clustering algorithms.

2. Formulation of the problem

There is a molecule of the SARS-CoV-2 virus and a set of antibodies, which consists of 43 elements. The task is to attach only two antibodies to the given virus molecule. The antibodies can be either different or the same (to distinguish them, one of them is marked with "*").

For simplicity, we will assume that the experiment happens in 2D, not 3D. Antibodies are two circles of approximately the same size with a small "beak" for interaction with the virus. Antibodies

IDDM 2022: 5th International Conference on Informatics & Data-Driven Medicine, November 18-20, 2022, Lyon, France;
EMAIL: sashko.zel2000@gmail.com (OZ); vitaliy.horlatch@lnu.edu.ua (VH); lebedin@xema.fi (YuL); p.yaryna@gmail.com (YaP).
ORCID: 0000-0003-1247-7511 (OZ); 0000-0001-5401-1731 (VH); 0000-0003-4250-4322 (YuL); 0000-0003-4834-9597 (YaP).


attach to the virus molecule, which is represented as a smaller circle. A schematic representation of this process can be seen in Figure 1.

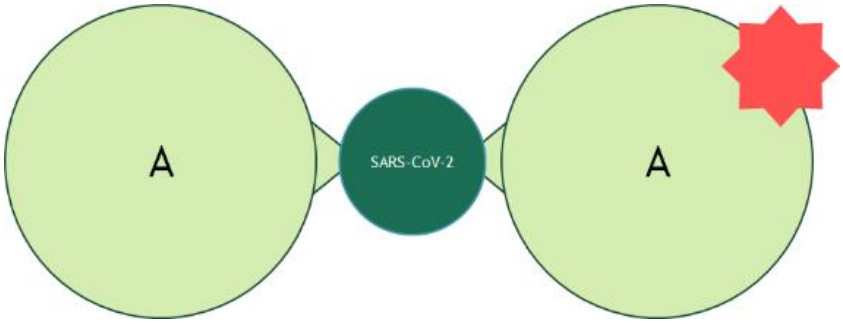

Figure 1: A schematic model of the attachment of antibodies to a viral molecule

In the case of the considered problem, the molar weight of the SARS-CoV-2 molecule is 45 kDa and the molar weight of the antibodies is 180 kDa. Since there is a need to attach two antibodies to one virus molecule, the main task is to find two antibodies (they can be identical) that are located at an optimal distance from each other. This means that they cannot overlap or be too close to each other because in such a case they start to compete and one of them cannot be attached.

During the manual experiments, it was discovered that this problem mostly can be solved by dividing the list of antibodies into groups according to how much they interfere with each other, or in other words, whether they can attach to the virus in the same region. If two antibodies belong to different groups, there is a very high probability that they will bind in different areas and interact better than if they were from the same group. However, in some cases, antibodies still will not be able to attach to the virus molecule.

Data from the experiment are presented in the form of a table, where each cell is the cross-inhibition index of the labeled antibody (from the column) and unlabeled (from the row). In the row marked as "blank", the maximum values of the cross-inhibition index for the corresponding labeled antibody are given.

3. Solutions for the problem

Since the dataset elements grouping problem is considered to be a problem of clustering, it was decided to apply one of the most popular types of clustering – the hierarchical algorithms, namely its' agglomerative subspecies. There were chosen several linkage methods [1]:

- Ward linkage – the increase in variance for the cluster being merged
- Complete linkage – the maximum distance between elements of each cluster
- Average linkage – the mean distance between elements of each cluster
- Single linkage – the minimum distance between elements of each cluster

In addition, it was decided to use the simplest Euclidean distance (1) as a metric

$$d(a,b) = \sqrt{\sum_i (a_i - b_i)^2}, \tag{1}$$

Before applying any algorithm, equation (2) was applied to each cell except the "blank" row.

$$cell_{i,j} = \frac{-\left(cell_{i,j} - blank_j\right)}{blank_j}, \tag{2}$$

The new values represent the percentage ratio between the value in the cell and the maximum value for the corresponding column. The new values are in the range of 0 to 1.

To develop an application for solving the described problem, the Python programming language was used. In particular, the “pandas” library was used to work with data and the “scikit-learn” library was used for clustering [4].

4. Results

As we can see at table 1 in the end, we want to get 11 groups (or 7 large groups) with different number of antibodies in each of them. From the experiment it is known that the best interaction will be between antibodies from the group 3B (X155, X41, X213, X32) and 4A (NP3706) or 4B (X211)

Table 1

Expected result

1A	1B	1B/2	2	2B/3	3A	3B	4A	4B	4C	5
NP1501	X190	NP1512	*NP1502*	NP1528	X202	X32	NP3706	X211	X215	X220
NP1514	NP1526	NP1521	*NP1503*		X218	X41				X275
NP1516	X200		*NP1508*		NP1518	X155				
NP1517	X201		*NP1510*		NP1527	X212				
NP1507			*NP1520*			X213				
			NP1522			X217				
			NP1525			X223				
			X221			X224				
			X271			X233				
			NP3701			NP1524				
			NP3708			NP3715				

As a result, we got 4 outputs for each linkage method. First, consider the result of using Ward linkage with distance threshold equal to 1.5.

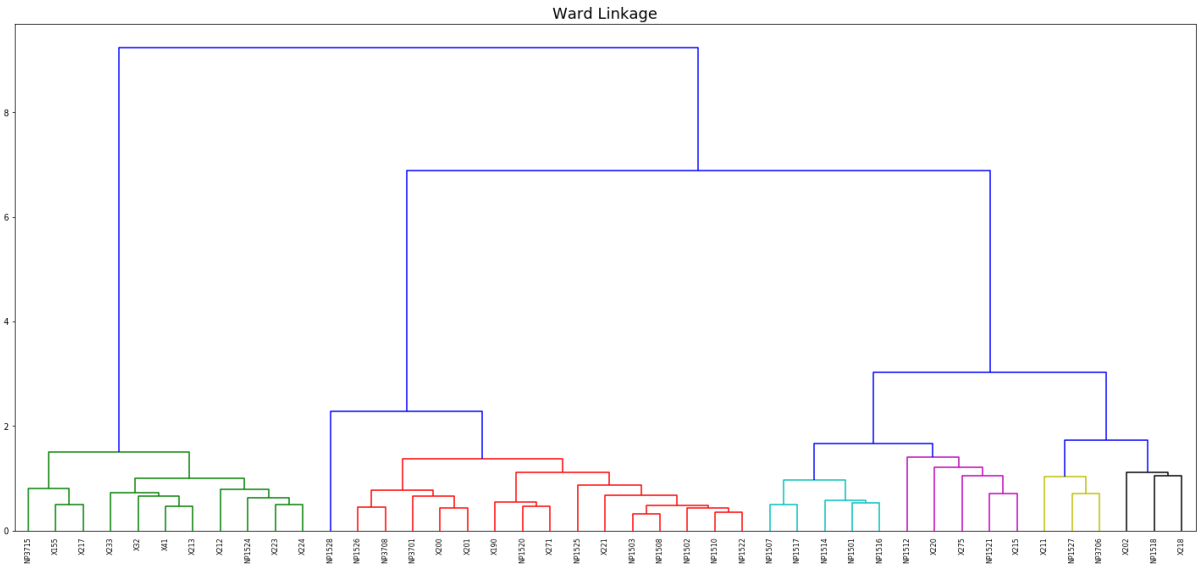

Figure 2: Dendrogram for agglomerative clustering with Ward linkage

The dendrogram (Figure 2) clearly shows that 7 clusters were identified as a result of the algorithm. In Table 2 we see the result of clustering where each column contains a list of antibodies that belong to the cluster.

It is clearly seen that cluster number 1 matches group 3B, cluster 5 completely matches group 2B/3 and cluster 7 matches group 1A (they are marked in green). Also, cluster 3 combines groups 1B and 2, cluster 4 corresponds to group 3A without the element NP1527, which is in cluster 6, which also contains groups 4A and 4B (they are marked in yellow and orange).

Table 2

Result for agglomerative clustering with Ward linkage

1	2	3	4	5	6	7
X41	NP1521	X200	X202	NP1528	NP1527	NP1517
X32	X275	X190	NP1518		X211	NP1514
X233	X215	X221	X218		NP3706	NP1516
X224	NP1512	X201				NP1507
X223	X220	X271				NP1501
X217		NP3708				
X213		NP3701				
X212		NP1526				
X155		NP1525				
NP3715		NP1522				
NP1524		NP1520				
		NP1502				
		NP1503				
		NP1510				
		NP1508				

Second, consider the result of using complete linkage with distance threshold equal to 1.2.

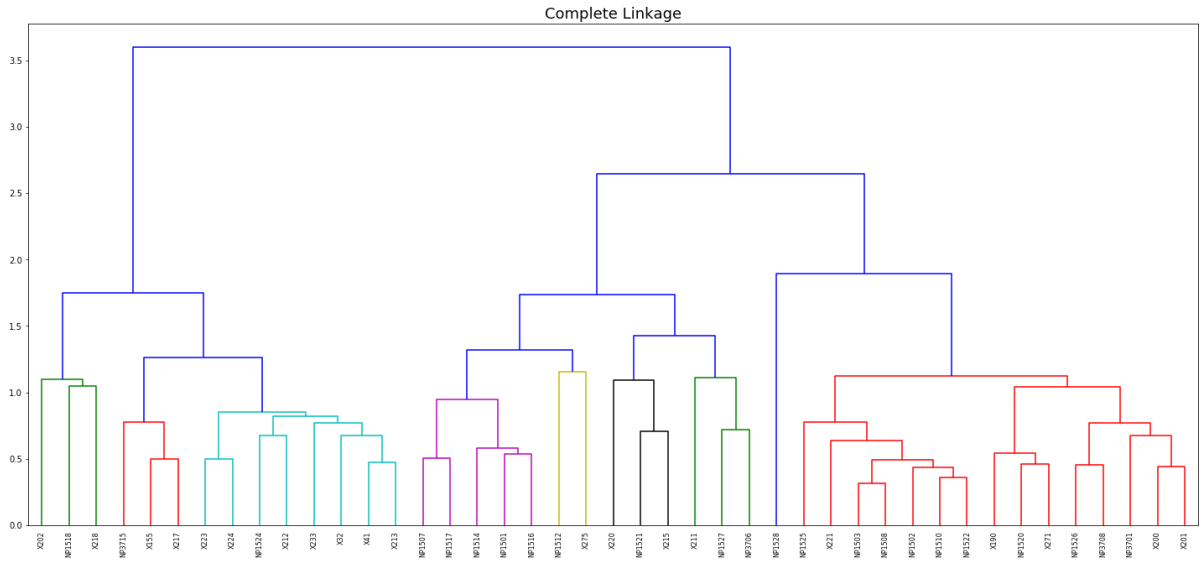

Figure 3: Dendrogram for agglomerative clustering with complete linkage

The dendrogram (Figure 3) clearly shows that 9 clusters were identified as a result of the algorithm. In Table 3 we see the result of clustering.

It is clearly seen that cluster number 4 matches group 1A and cluster 8 completely matches group 2B/3 (they are marked in green). Also, cluster 2 combines groups 1B and 2, cluster 5 corresponds

to group 3A without the element NP1527, which is in cluster 3, which also contains groups 4A and 4B, in addition cluster 6 and cluster 9 contains the elements from group 3B (they are marked in yellow and orange).

Table 3

Result for agglomerative clustering with complete linkage

1	2	3	4	5	6	7	8	9
X275	X221	X211	NP1507	X202	X213	NP1521	NP1528	X217
NP1512	X201	NP1527	NP1501	NP1518	X32	X215		X155
	X200	NP3706	NP1516	X218	X41	X220		NP3715
	X190		NP1517		X233			
	X271		NP1514		X224			
	NP3701				NP1524			
	NP1526				X212			
	NP1525				X223			
	NP1522							
	NP3708							
	NP1502							
	NP1503							
	NP1520							
	NP1508							
	NP1510							

Third, consider the result of using average linkage with distance threshold equal to 0.97.

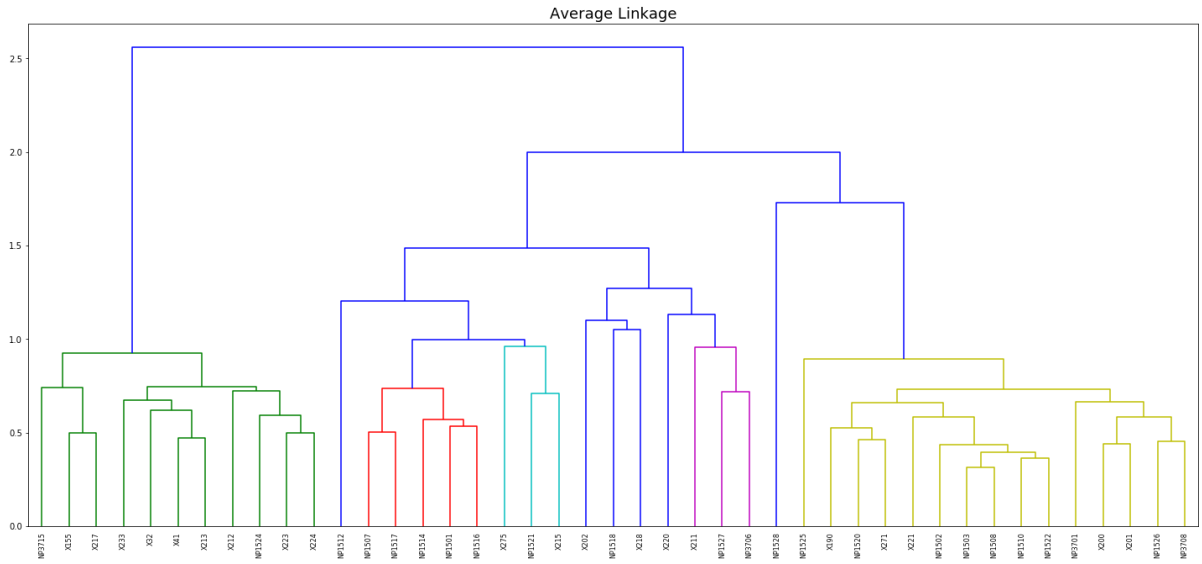

Figure 4: Dendrogram for agglomerative clustering with average linkage

The dendrogram (Figure 4) clearly shows that 11 clusters were identified as a result of the algorithm. In Table 4 we see the result of clustering.

It is clearly seen that cluster number 2 matches group 3B, cluster 5 completely matches group 1A and cluster 8 matches group 2B/3 (they are marked in green). Also, cluster 6 combines groups 1B and 2, clusters 9, 10 and 11 corresponds to group 3A without the element NP1527, which is in cluster 6, which also contains groups 4A and 4B (they are marked in yellow and orange).

Table 4

Result for agglomerative clustering with average linkage

1	2	3	4	5	6	7	8	9, 10, 11		
X275	NP3715	X211	X220	NP1501	NP1502	NP1512	NP1528	X202	NP1518	X218
NP1521	NP1524	NP3706		NP1517	NP1503					
X215	X32	NP1527		NP1507	X221					
	X41			NP1514	NP1508					
	X155			NP1516	NP1510					
	X212				NP3701					
	X213				X200					
	X217				X190					
	X223				NP1520					
	X224				NP1522					
	X233				NP1525					
					X271					
					NP1526					
					X201					
					NP3708					

Finally, consider the result of using single linkage with distance threshold equal to 0.75.

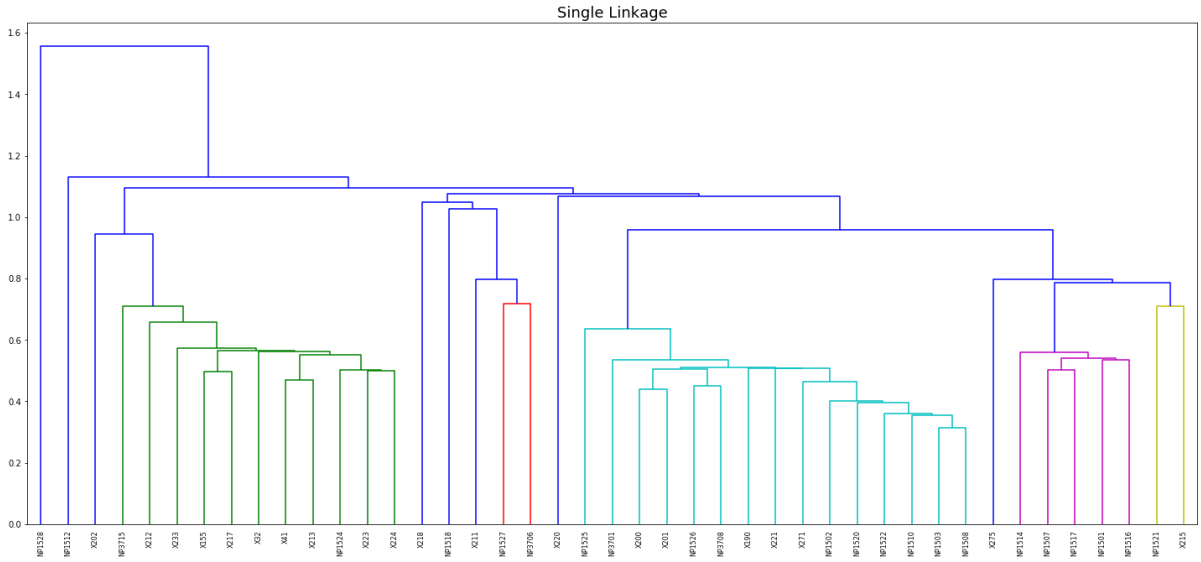

Figure 5: Dendrogram for agglomerative clustering with single linkage

The dendrogram (Figure 5) clearly shows that 13 clusters were identified as an output of the algorithm. In Table 5 we see the result of clustering.

It is clearly seen that cluster number 2 matches group 3B, cluster 3 completely matches group 1A and cluster 6 matches group 1A and cluster 13 matches group 4B (they are marked in green). Also, cluster 4 combines groups 1B and 2, cluster 6 and 11 contains items from group 5 (they are marked in yellow and orange).

Table 5
Result for agglomerative clustering with single linkage

1	2	3	4	5	6	7	8	9	10	11	12	13
NP1527	X213	NP1517	X201	NP1521	X220	X218	NP1518	NP1528	X202	X275	NP1512	X211
NP3706	X32	NP1514	X200	X215								
	X41	NP1507	X221									
	X155	NP1516	X190									
	NP1524	NP1501	NP3708									
	X212		NP3701									
	NP3715		NP1502									
	X217		NP1526									
	X223		NP1525									
	X224		NP1522									
	X233		NP1503									
			NP1508									
			NP1520									
			X271									
			NP1510									

5. Conclusion

As a metric of accuracy, the total amount of elements in the clusters, which fully correspond to the expected result, was taken. Based on this metric, it is obvious that the algorithm, which used a single linkage method, gives the best result. However, the algorithms, which used ward linkage and average linkage methods, are not much worse. Surprisingly, the algorithm, which used the complete linkage method is the worst.

Even though the amount of data may seem to be small (40x30 matrix), the developed application does the amount of work, that would take a person several days to complete, in a short time (1-2 minutes). Moreover, as the sample data size increases, the amount of time it takes for the computer to execute the algorithm will remain small compared to the time it would take a person to perform the same task.

In conclusion, hierarchical clustering methods have shown themselves to be quite suitable for a given problem. However, they do not take into account the order in which it forms the clusters (the order of the clusters is not the same as the order of the groups in the expected result) yet, but it is also a key aspect of this problem.

6. References

[1] F. Nielsen, Introduction to HPC with MPI for Data Science, Chapter 8: Hierarchical Clustering, Springer, Switzerland, 2016, 195–211. URL: https://www.researchgate.net/publication/314700681_Hierarchical_Clustering. http://dx.doi.org/10.1007/978-3-319-21903-5_8.

[2] O. Zelinskyi, V. Horlatch, Yu. Lebedin, Development of antibody clusterization system based on coefficient of cross-inhibition, International Student Scientific Conference of Applied Mathematics and Computer Science (ISSCAMCS – 2022), May 5-6, 2022, Lviv, Ukraine, 8-12, URL: https://ami.lnu.edu.ua/wp-content/uploads/2022/05/ISSCAMCS-2022.pdf

[3] WHO Coronavirus (COVID-19) Dashboard, 28 September 2022, URL: https://covid19.who.int/

[4] Scikit-learn documentation, AgglomerativeClustering, 2022, URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html?highlight=ag#sklearn.cluster.AgglomerativeClustering