U2151556

Task-1

Question-1:

Following are the start and end positions of the nucleotides encoding the S-proteins of the three sequences(per the approximate length in the question):

- Wuhan(*NC_045512.2*) → start_pos = 21535; end_pos = 25384
- Omicron(*OM095411.1*) → start_pos = 21487; end_pos = 25327
- Hiroshima(*OL638173.1*) → start_pos = 21510; end_pos = 25353

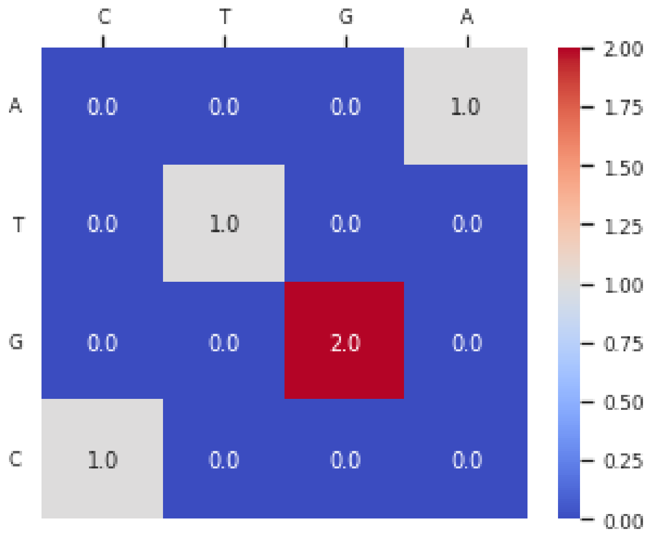
They are cross-validated by checking the start and end codon of the sequences, as they all have "ATG" as start codon and "TAA" as stop codon.

Question-2:

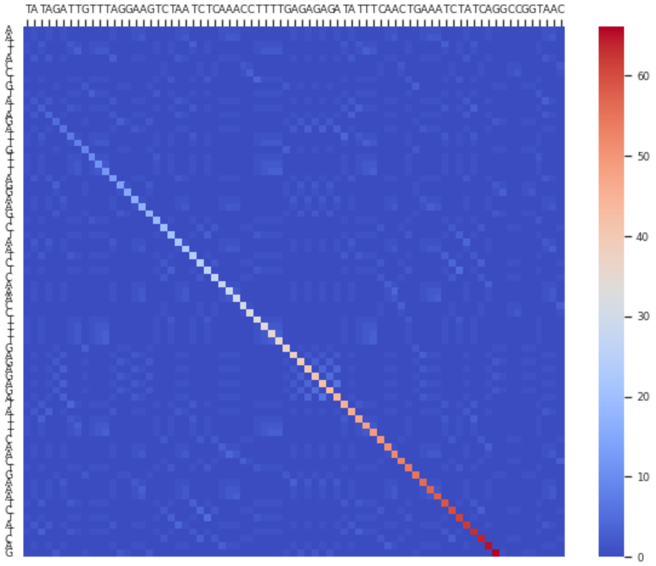
Approach - I have developed a logic using 2-D numpy arrays to create the dot plots for all the combinations of sequences in the given RBD region, and have shown them using *matplotlib* method *Imshow()*.

Steps:- given any two sequences, say Seq1 & Seq2.

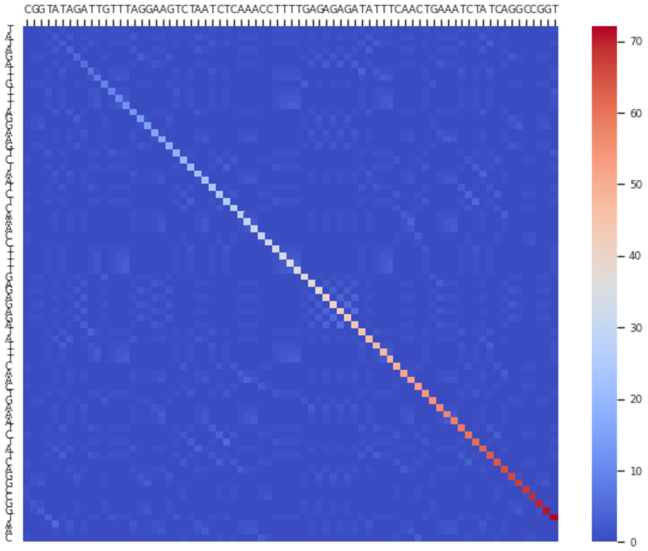
- The function creates a zero 2-D array of shape **"Seq1 x Seq2" ,** say **M.**
- Iterates over all the rows and columns, and updates the value at any index (**i, j**), by adding **1** to the immediate north-west diagonal value, if the nucleotides are the same at that particular index (**i**, **j**).
- For Example: Seq1 = "ATGC" and Seq2 = "CTGA", then the matrix would look like:



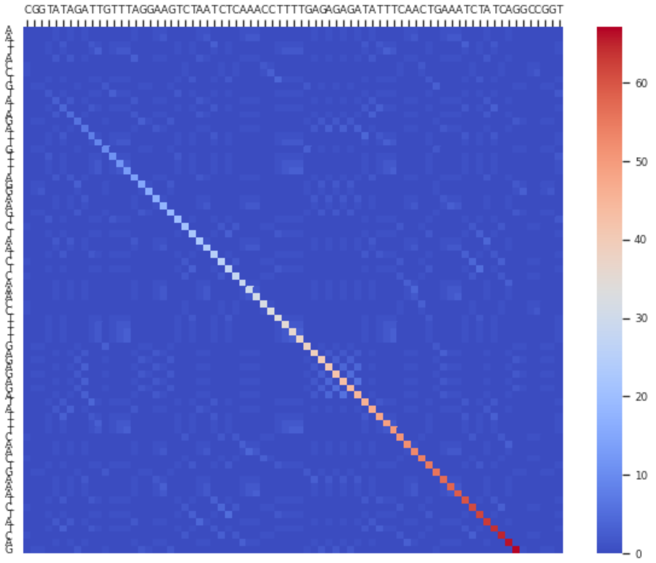
Following are the result of the dot plots of the pairwise combinations of all the sequences:



Wuhan-Omicron(run length - 66)



Omicron-Hiroshima(run length - 72)

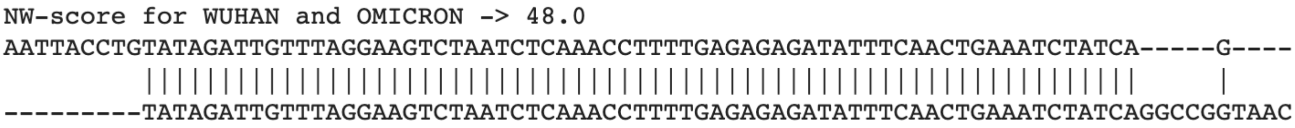


Wuhan-Hiroshima(run length - 67)

Question-3:

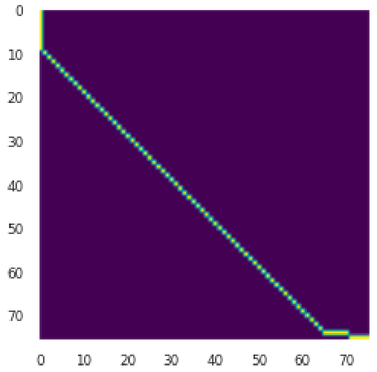
The Needleman-Wunch algorithm is implemented to the given RBD region using a scoring function in the form of a matrix. Two functions are written to do so, (1). **nwa** - which outputs the final matrix (2). **backtr** - utilises the matrix and the sequences to generate the alignment and to plot the backtrack.

Wuhan and Omicron:

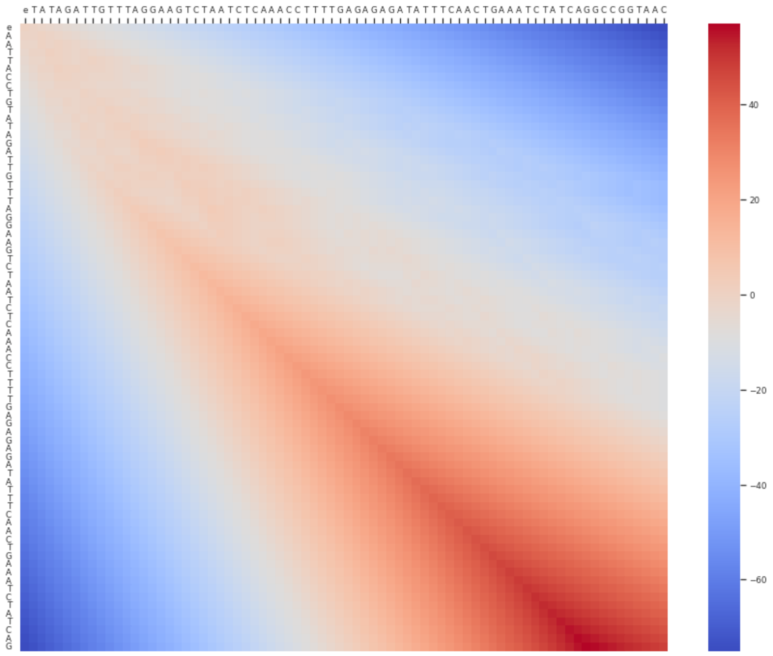


```
NW-score for WUHAN and OMICRON -> 48.0
AATTACCTGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCA-----G----
         |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |    
---------TATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGTAAC
```

Score and the Alignment



Backtrack

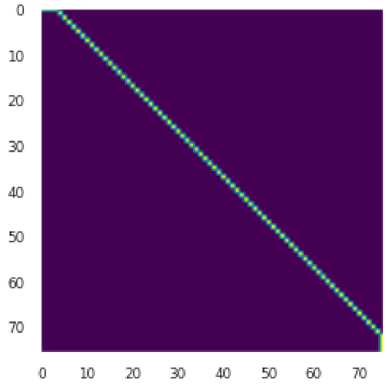


NW Matrix

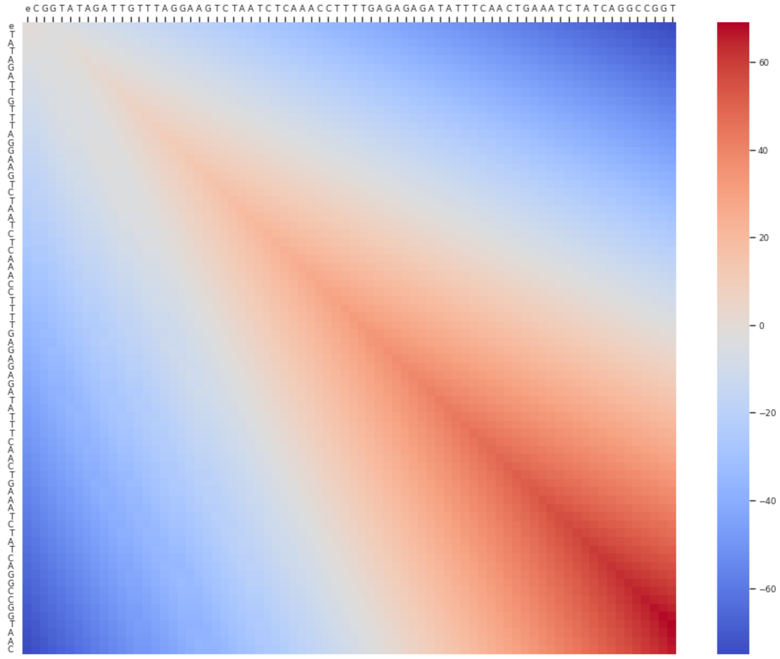
Omicron and Hiroshima:

```
NW-score for OMICRON and HIROSHIMA -> 66.0
---TATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGTAAC
   ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
CGGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGT---
```

Score and the Alignment



Backtrack

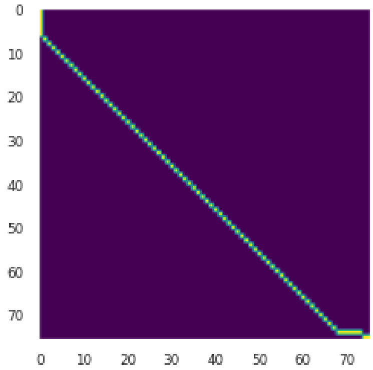


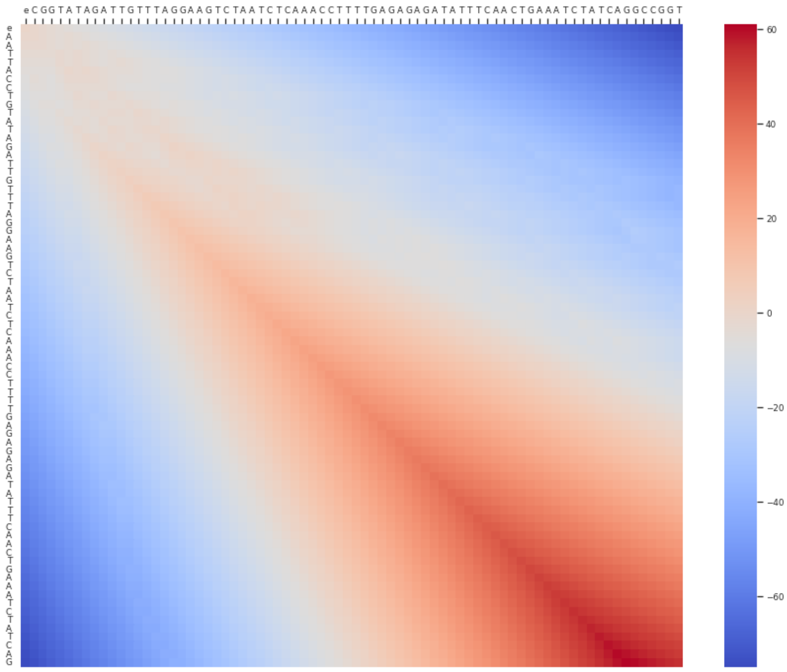
NW Matrix

Wuhan and Hiroshima:

```
NW-score for WUHAN and HIROSHIMA -> 55.0
AATTACCTGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCA-----G-
      | ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |
------CGGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGT
```

Score and the Alignment





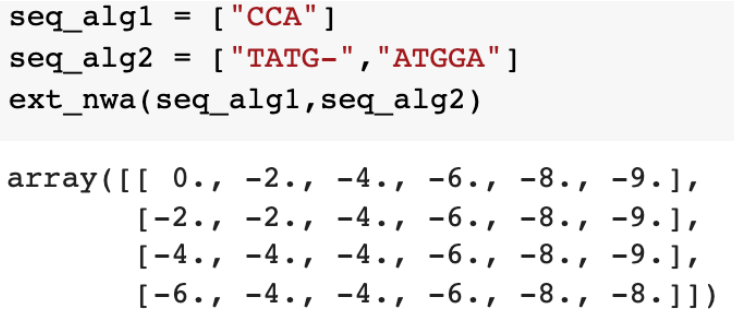
Backtrack

NW Matrix

Question-4:

Both functions in Question-3 are modified “**ext_nwa**”, “**backtr_ext**” to cater the implementation of the same algorithm for generating alignment of alignments and showing corresponding results.

A demonstration of its implementation on the example discussed in the lecture. The matrix aligns with the matrix computed by the Professor.



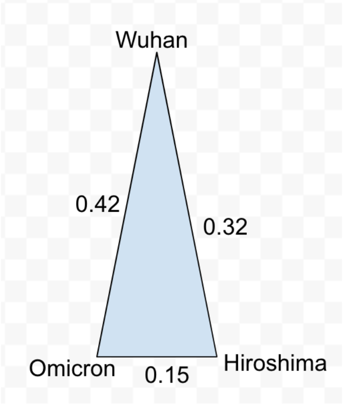
```
seq_alg1 = ["CCA"]
seq_alg2 = ["TATG-","ATGGA"]
ext_nwa(seq_alg1,seq_alg2)
```

```
array([[ 0., -2., -4., -6., -8., -9.],
       [-2., -2., -4., -6., -8., -9.],
       [-4., -4., -4., -6., -8., -9.],
       [-6., -4., -4., -6., -8., -8.]])
```

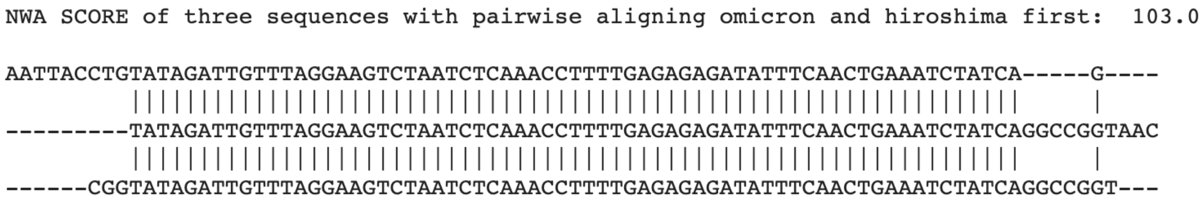
Question-5:

Two approaches were devised based on the exploration of Question-3 and Question-4 to understand which two sequences should get aligned first.

Approach-1 :- Using NWA written for Question-3, we computed normalised distances (using length of alignments) between all the combinations to identify which two sequences are closer and farthest from the third one.



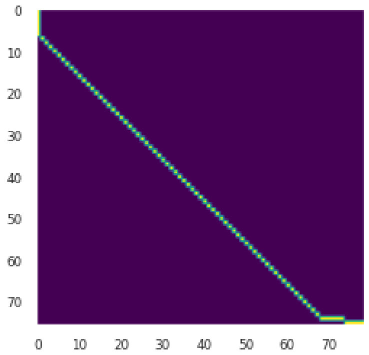
According to this approach, it turns out that Omicron and Hiroshima should get aligned first. Following are the results of alignment of alignments for the same.



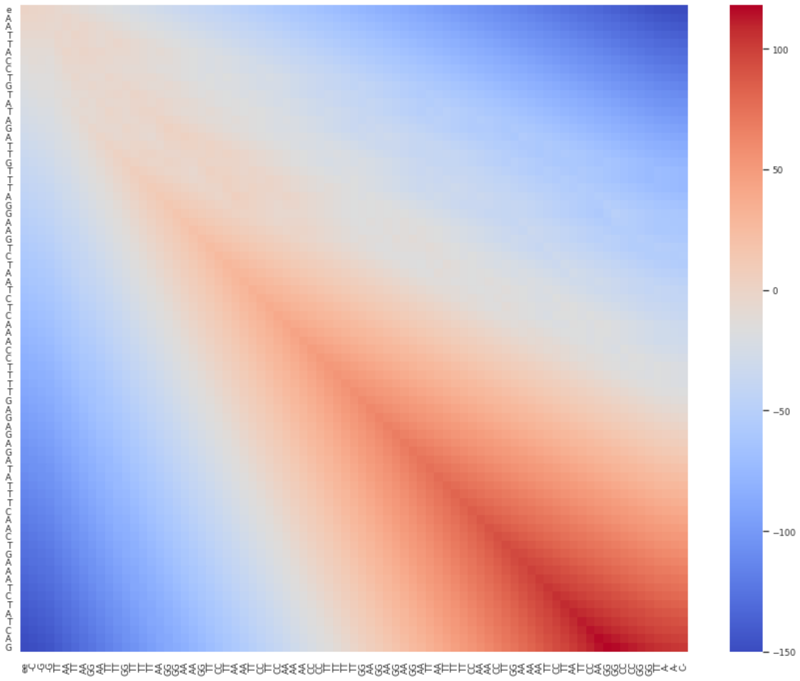
```
NWA SCORE of three sequences with pairwise aligning omicron and hiroshima first:  103.0

AATTACCTGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCA-----G----
         |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |
---------TATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGTAAC
         |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |
------CGGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGT---
```

Score and the alignment

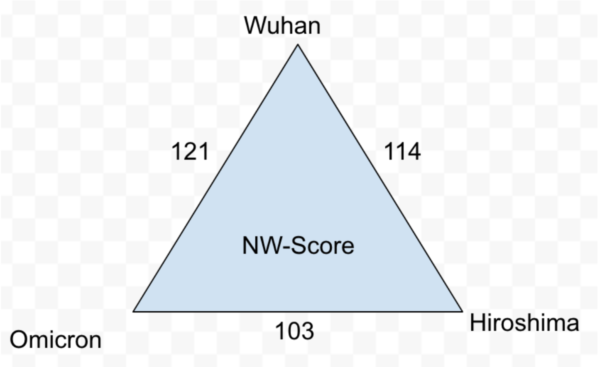


Backtrack

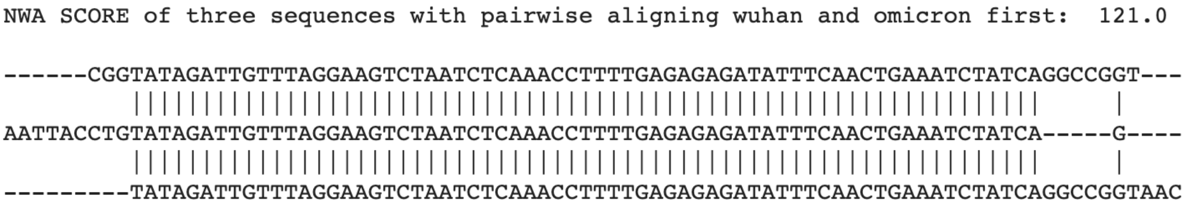


NW matrix

Approach-2 :- Computed NW-score for alignment of alignments for all the three combinations(in which any two will get pairwise aligned first) of sequences. In the triangle below, each side represents the NW-score(alignment of alignments) when the vertices containing the side are pairwise aligned first.



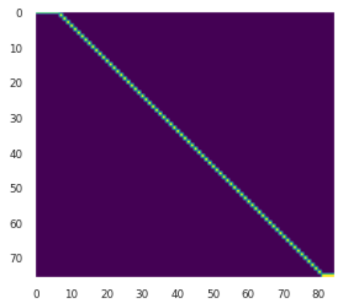
It shows that wuhan and omicron should be aligned first in order to get the highest NW-score. Following are the results for the same.



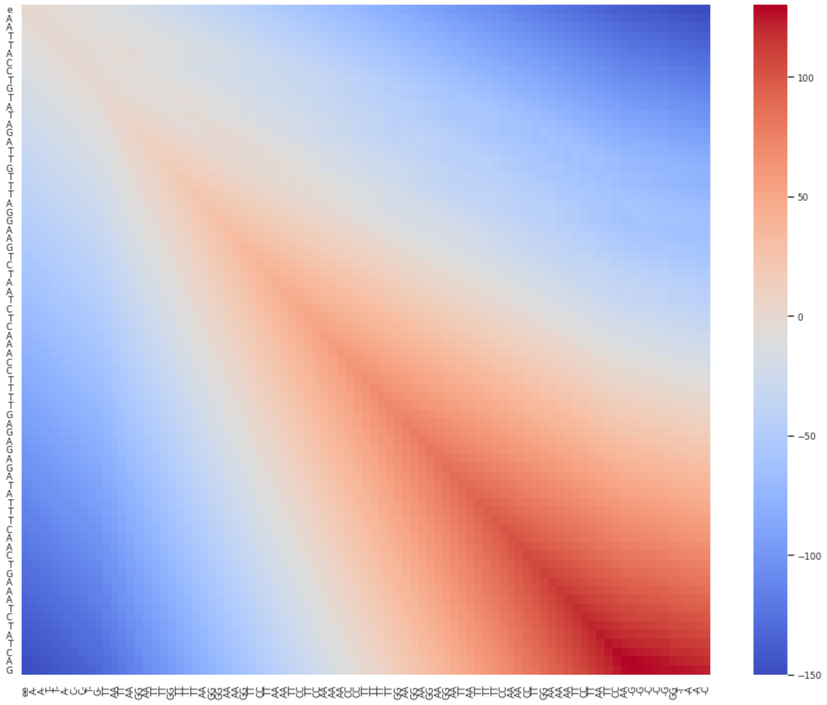
```
NWA SCORE of three sequences with pairwise aligning wuhan and omicron first:  121.0

------CGGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGT---
         |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |
AATTACCTGTATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCA-----G----
         |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||     |
---------TATAGATTGTTTAGGAAGTCTAATCTCAAACCTTTTGAGAGAGATATTTCAACTGAAATCTATCAGGCCGGTAAC
```

Score and the alignment



Backtrack

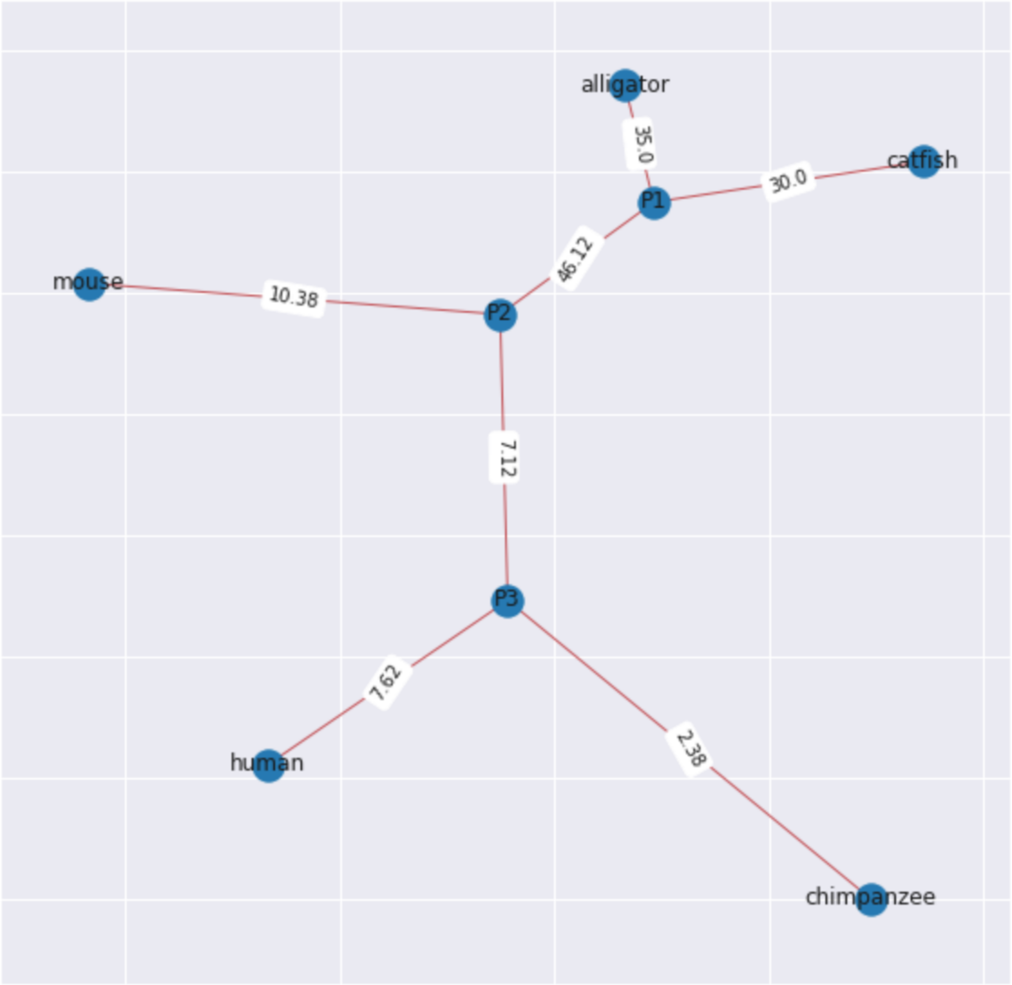


NW matrix

Task-2

Distance matrix is computed using normalised scores of the pairwise alignments for the combinations of all sequences. A function “**NJM_pair**” takes the distance matrix as input and iteratively implements the neighbour joining algorithm until the size of the distance matrix reduces to 2x2 . The function returns a mapping dictionary where the key contains the information of pairs merged to a new node in each iteration, and the value gives the distance array of the new node from all the sequences.

This mapping dictionary with the use of *networkx* library utilised to represent the phylogenetic tree as a graph.


alligator
35.0
catfish
30.0
P1
46.12
mouse
10.38
P2
7.12
P3
7.62
2.38
human
chimpanzee