COMP3340 Assignment 1 part 2

By Andrew cox

Question 1

A).

Between certain variables there is indeed a strong correlation as found through the use of pearsons correlation.

B).

1. Students who raided their hands are more actively involved in study related works. ()

2. There is no apparent gender bias in topic/subject selection. ()

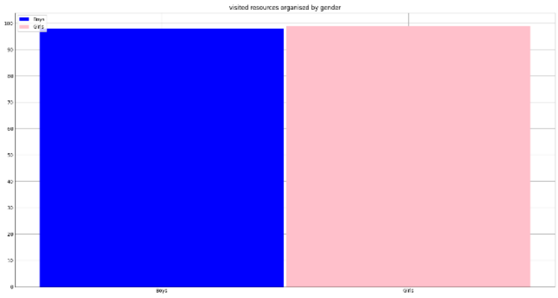
3. Girls seem to have better overall performance than boys. ()

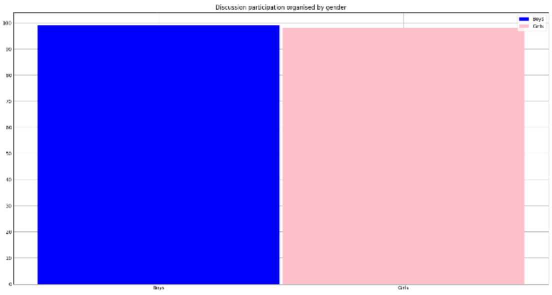
-

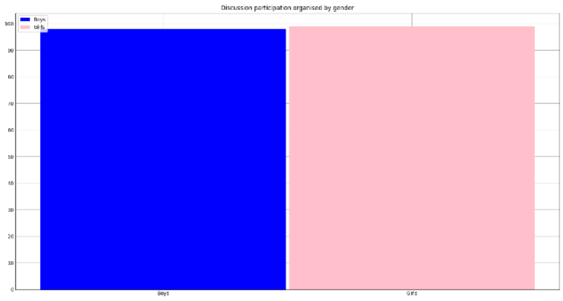
4. Boys are generally more open to: discussion, visiting resources, and raising hands. (F)

5. Those who participated more usually preform better. ()

Graphs



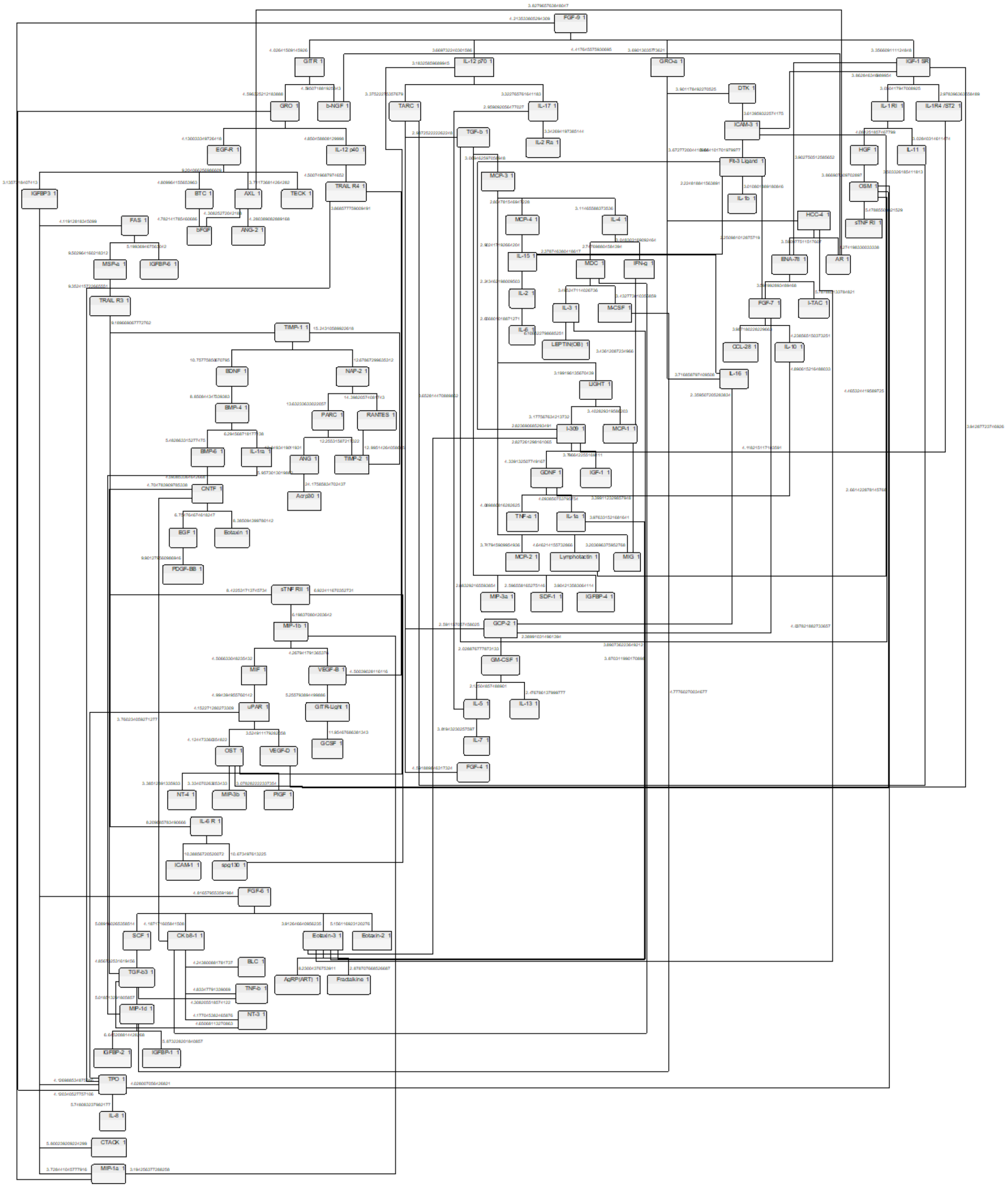




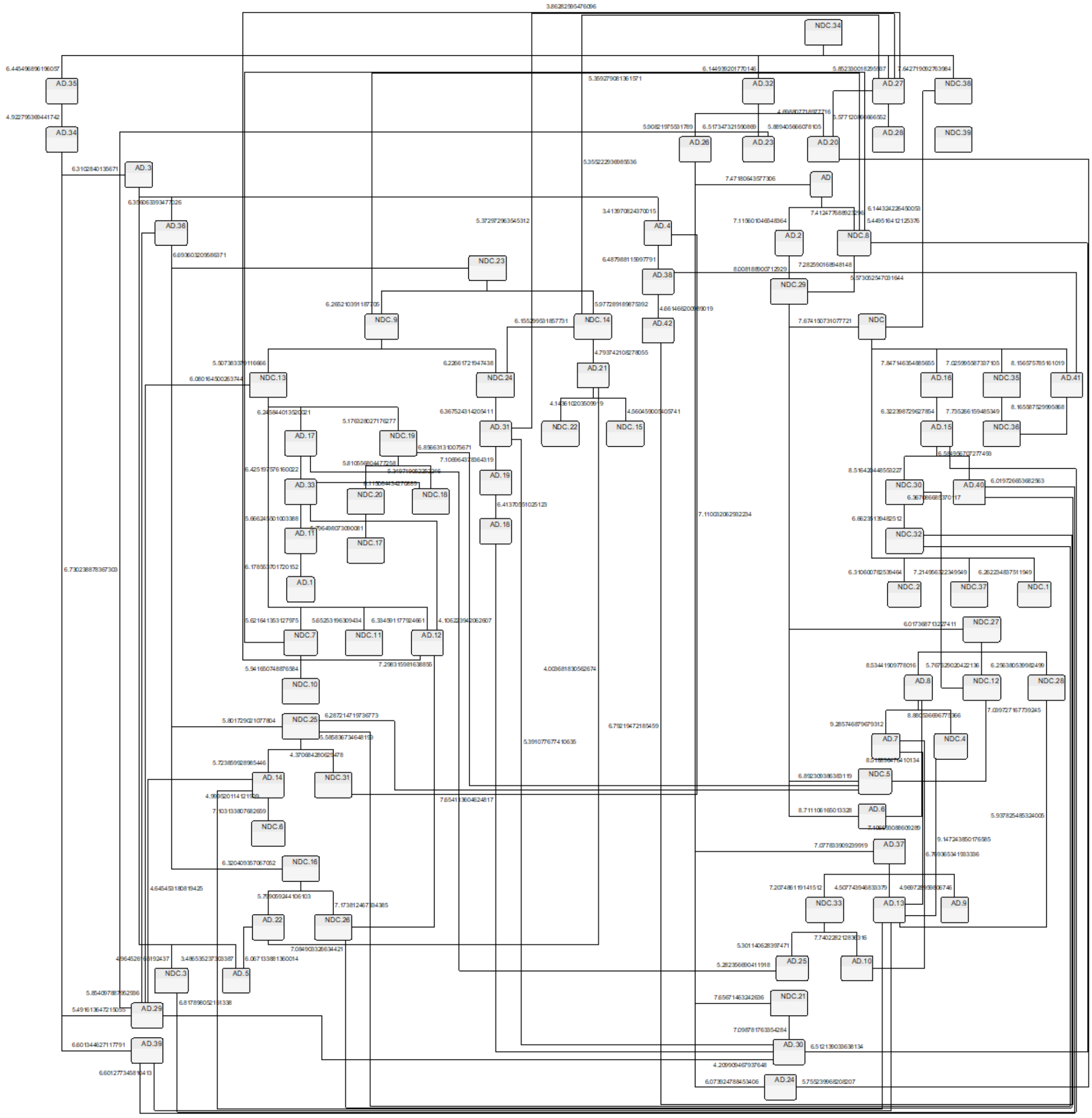
Question 2

Graphs

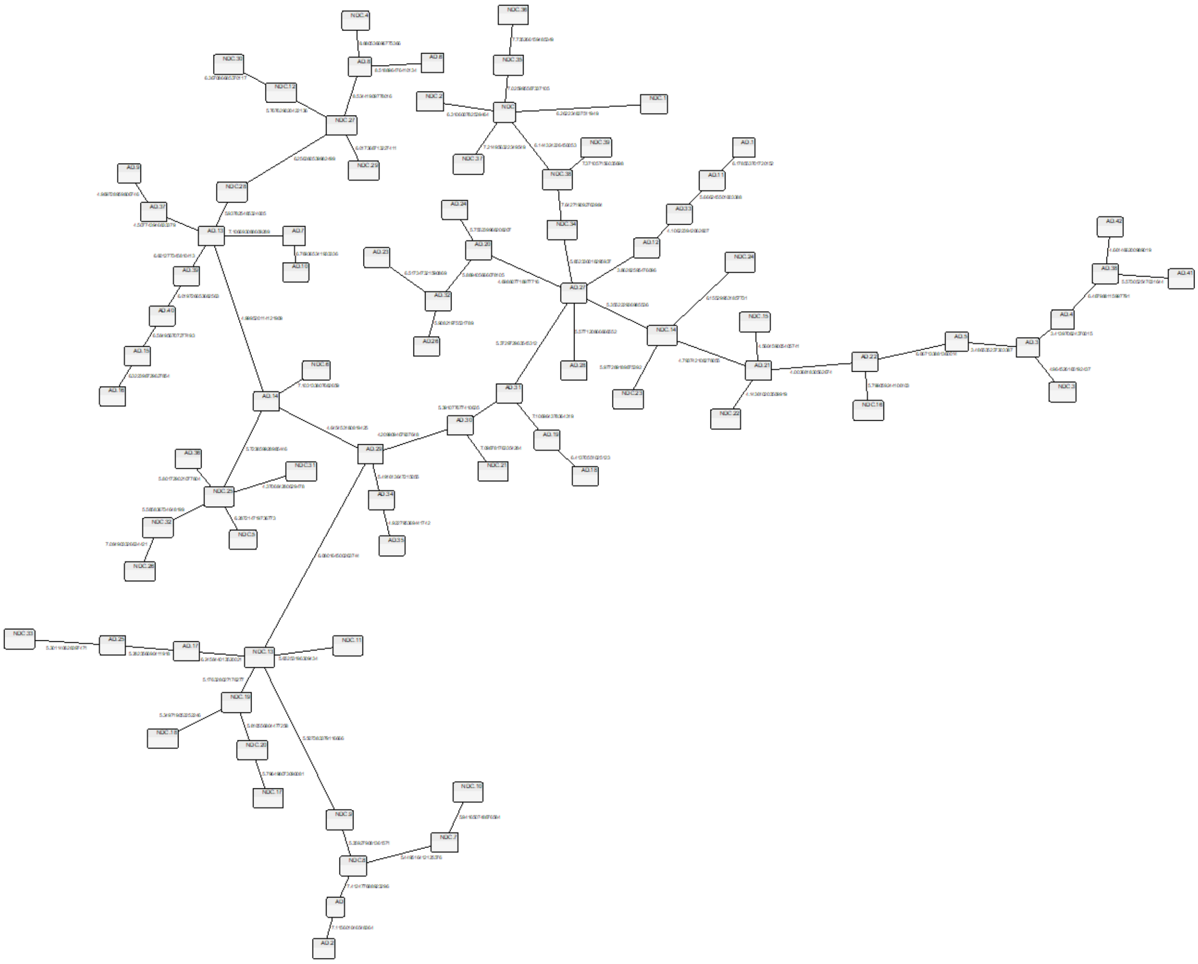
Proteins Relative neighbourhood graph



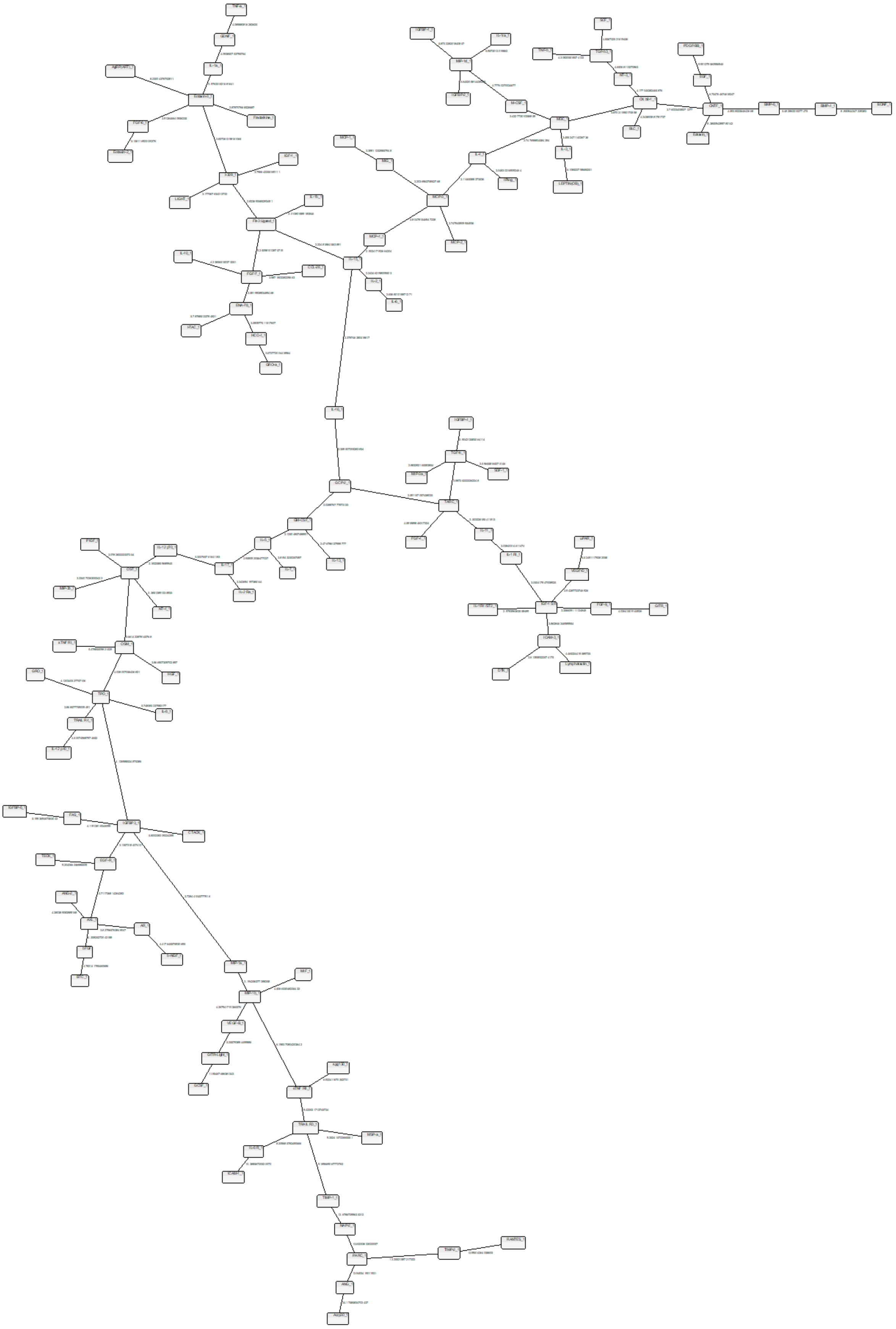
Samples Relative neighbourhood graph



Samples Minimum spanning tree.



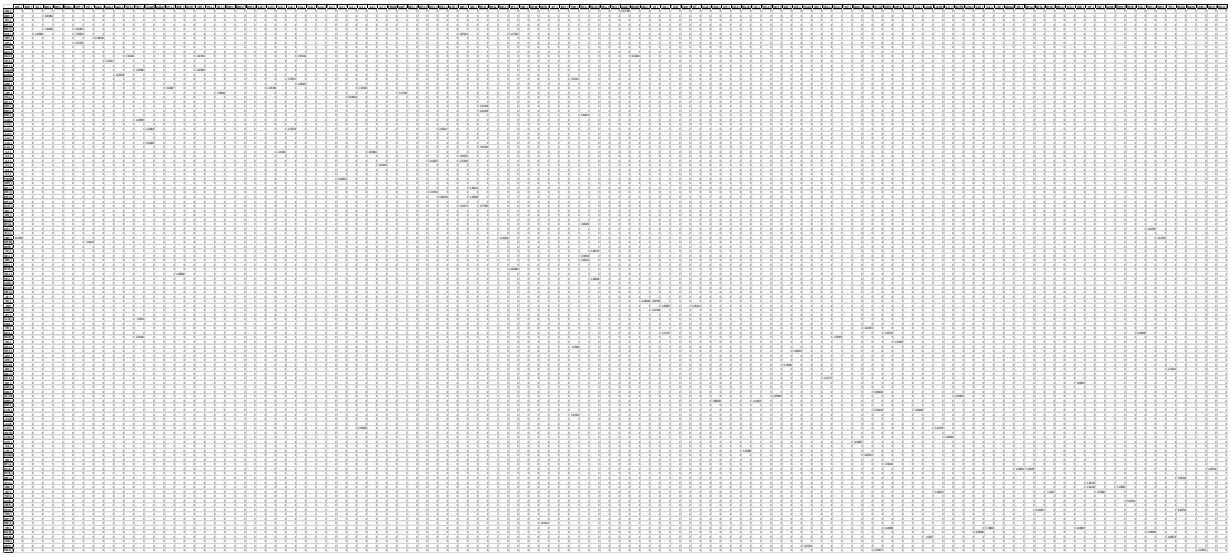
Proteins Minimum spanning tree.



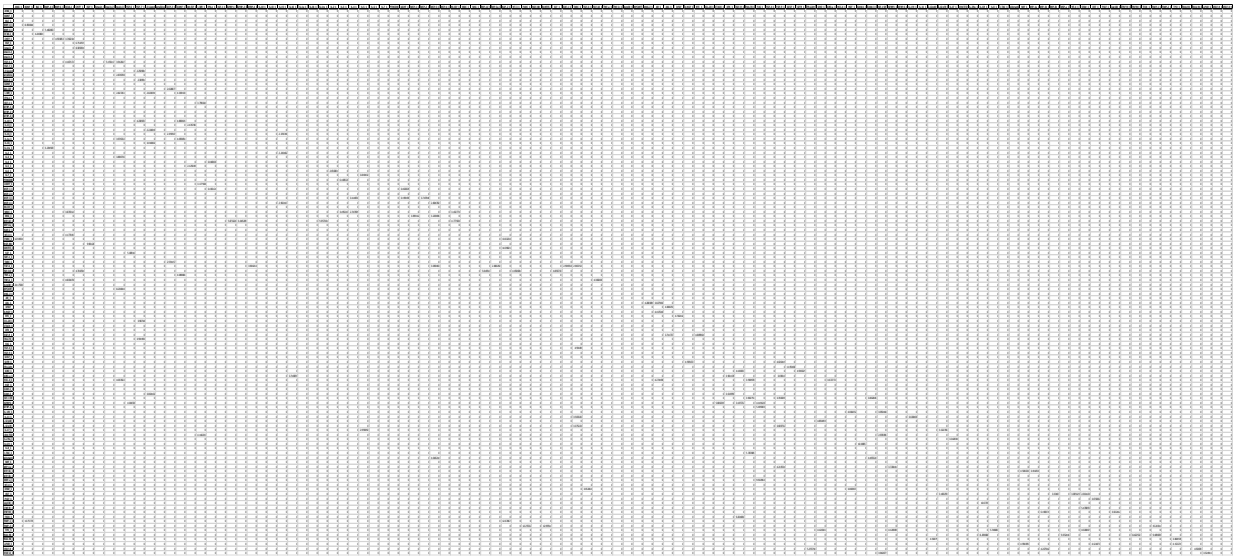
Matrix’s

Proteins MST & RNG

Minimum spanning tree

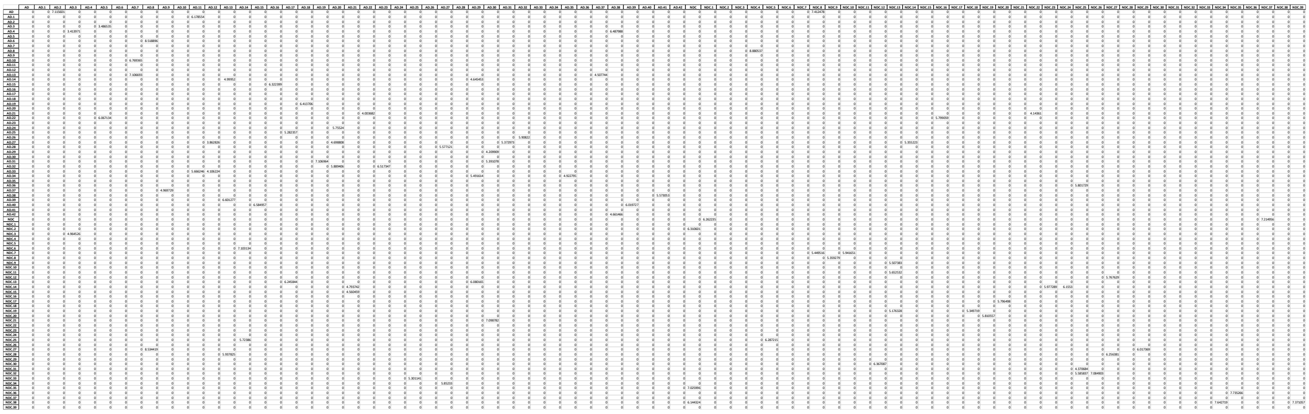


Relative Neighbourhood Graph

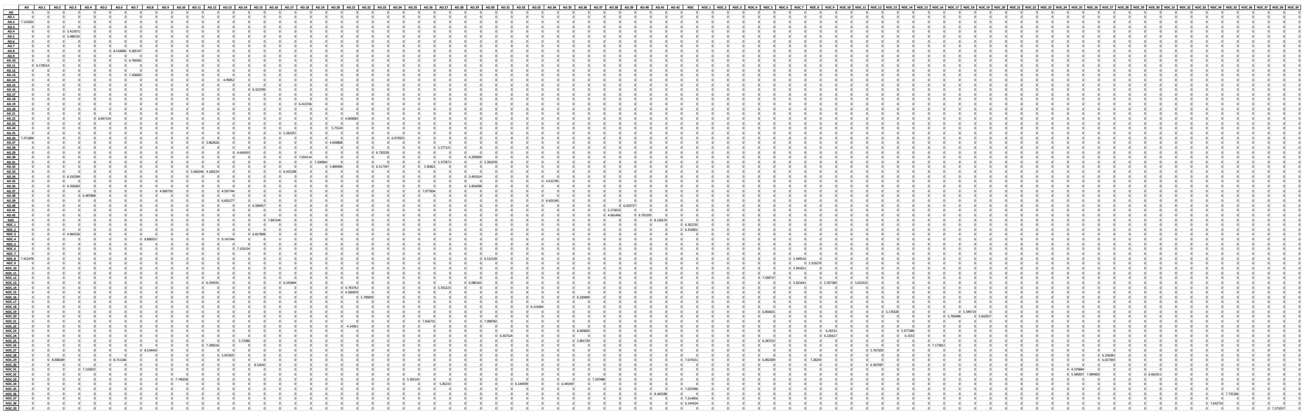


Samples MST & RNG

Minimum spanning tree



Relative neighbourhood graph



How was it done?

To generate the distance matrix I used the SCIPY euclidean() method. This checks the distance between two input arrays no matter of their contents. Where as with the hemming matrix it would only work properly for binary. I just used the same method I created in assignment 1 part 1 for question 1 retrofitted to generate a Euclidian matrix rather than a hemming matrix. For the samples I transposed the matrix using the numpy transpose() method.

To Actually generate the MST and RNG I used external libraries from SCIPY and relativeNeighborhoodGraph respectively. To generate the MST I parsed the distance matrix, index names and the export location to the generic method genMST() which used the SCIPY method minimum_spanning_tree() to generate a minimum spanning tree using Kruskal's algorithm. The minimum_spanning_tree() method would return a sparse array which was then to be converted into a normal array. And exported to an XLSX file using pandas.

To generate the RNG I used the same method as with he MST however with another step. The method returnRNG() returns a pandas data frame with duplicated lines, to fix this issue I created a method to loop through the matrix checking for duplicate lines (a line is duplicate if it appears in both x, y and y, x).

Question 3

Question 4

Question 5