Team members:

Ansuman D. Mohanty- 2016A7PS0043H
Syed Abid Abdullah - 2016A7PS0562H
Mridul Bhaskar - 2016A7PS0391H
Deepak Gupta - 2016A7PS0105H

Dataset Used:

Amino Acid Sequence

Pre-processing Done on the data:

In the k-means clustering algorithm, the code sometimes gave us an empty cluster because two of the data points were exactly similar. Hence we removed one of the duplicate data points. Later we modified the code to handle this exceptional case.

Formulas Used:

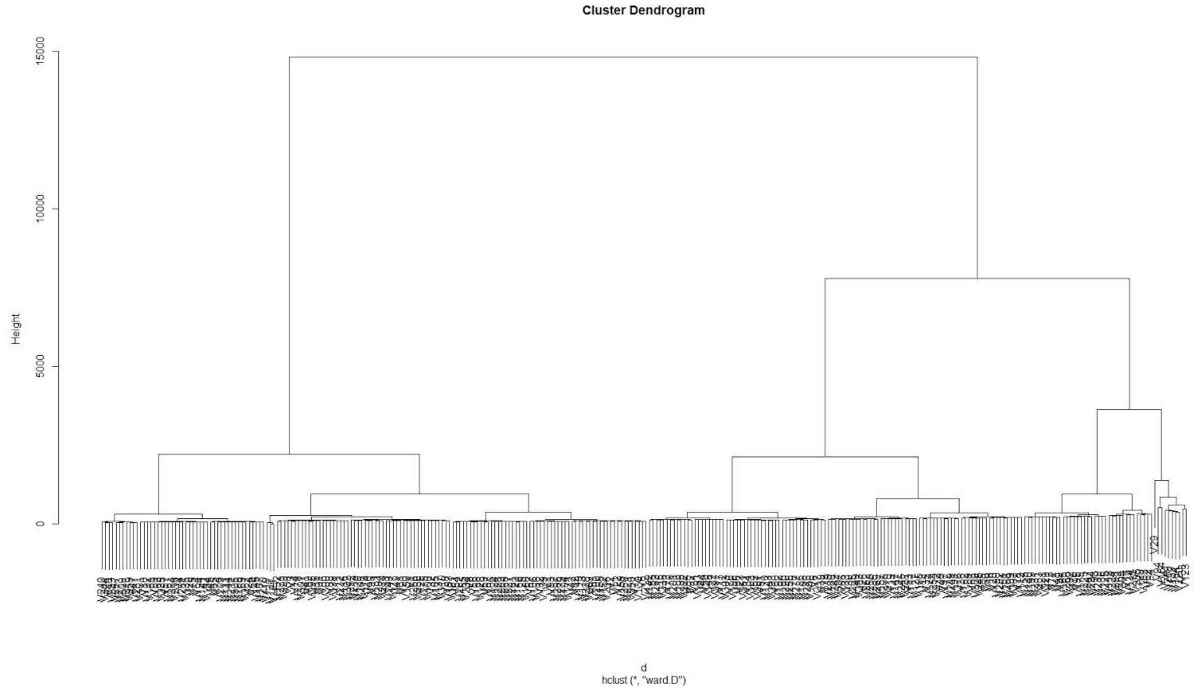
S.No.	Purpose	Formula
1.	**To calculate distance matrix**	$F_{i,j}=\min(F_{i-1,j-1}+S(A_i+B_j),F_{i-1,j}+d,F_{i,j-1}+d)$
2.	**To find convergence for k-means:** Since the centroid of cluster in our case didn't make sense, so we minimised the intra-cluster sum of squares.	$\text{cost}(C_1,\ldots,C_k;z_1,\ldots,z_k)=\sum_{j=1}^{k}\sum_{x\in C_j}\lVert x-z_j\rVert^2.$

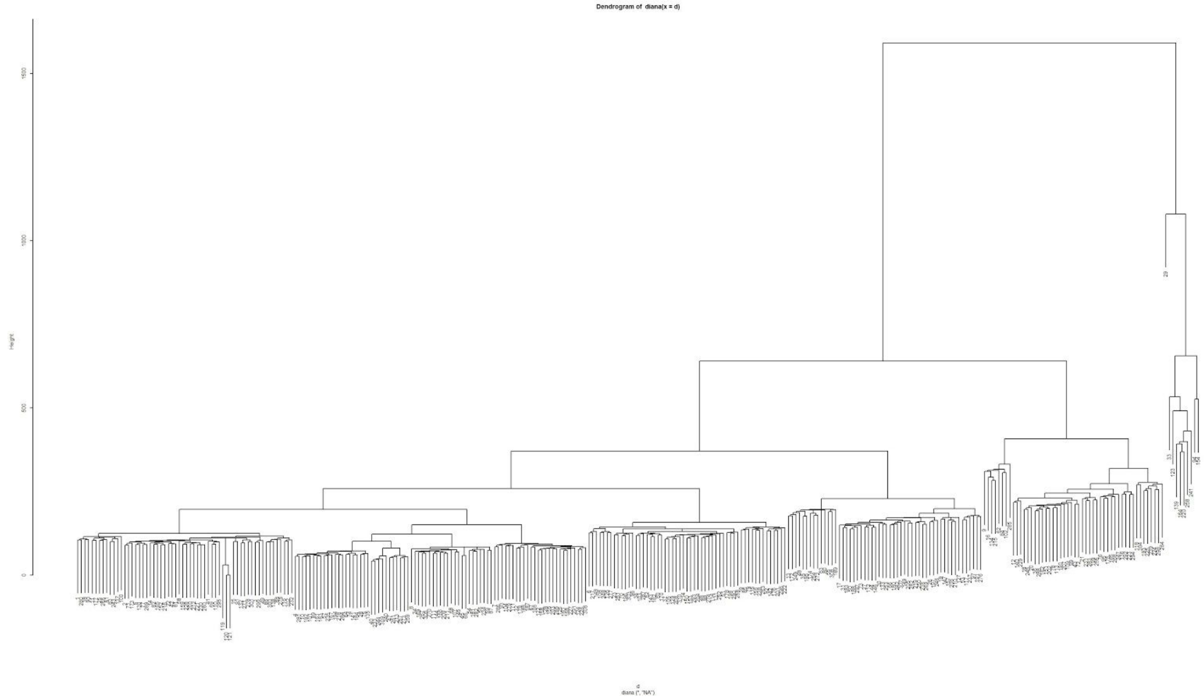
Linkage and distance metric used and the type of data it can cluster properly:

The global alignment between the amino acid sequences is stored in the form of a 2-D matrix which well help to cluster the points according to the proximity between two points based on the distance stored in matrix. A modification (of penalties) of **Needleman–Wunsch algorithm** was used to calculate the distances where we denoted the missing length by 0. The insertion/deletion were given a penalty of 2 and the the mismatch was given a penalty of 1. The distance metric used for agglomerative approach is min,max,av for divisive average is used and for k-means min is used.

Min : Can handle non elliptical shapes. But sensitive to noise and outliers.
Max: Less susceptible to noise and outliers. But it tends to break large clusters.
Av: Less susceptible to noise and outliers. But biased towards globular clusters.

<u>Comparison of dendrogram plot of top-down and bottom-up clustering:</u>





<u>The first figure shows the bottom-up and second one shows the top-down clustering.</u>
This dataset seems to have some amount of noise due to which the first splinter cluster is of small size.

MSS Value Comparison

k	k-means	Agglomerative - min	Agglomerative - max	Agglomerative-av	Divisive
1	61375	61375	61375	61375	61375
2	53016	59263	57756	53073	53732
4	47990	56269	56057	47909	42222
8	45097	52155	47363	44421	40802
16	42225	49835	42883	42275	41368
32	40124	42434	35948	38867	38417
64	33715	32647	28686	32841	32662
128	23992	19213	17243	22561	22023
256	6801	3087	3379	3910	4157

Comparison of k-means with Agglomerative:

The WSS value for both the k-means and Agglomerative is decreasing with the number of clusters. WSS for k-means is less for most of the values but for very small and very large clusters the value of Agglomerative is less.

Comparison of k-means with Divisive:

The WSS value for the k-means is decreasing with the number of clusters while it is increasing for k-values 8 to 16 for divisive and decreasing with k for all other values. WSS for Divisive decreases sharply for k-values 2 to 4. WSS for Divisive is less than that of k-means for all the k-values.

Comparison of Agglomerative with Divisive:

The WSS value for the Agglomerative is decreasing with the number of clusters while it is increasing for k-values 8 to 16 for divisive and decreasing with k for all other values. WSS for Divisive decreases sharply for k-values 2 to 4. The WSS for divisive is less than agglomerative till k=32 and thereafter the value for Agglomerative is lesser.