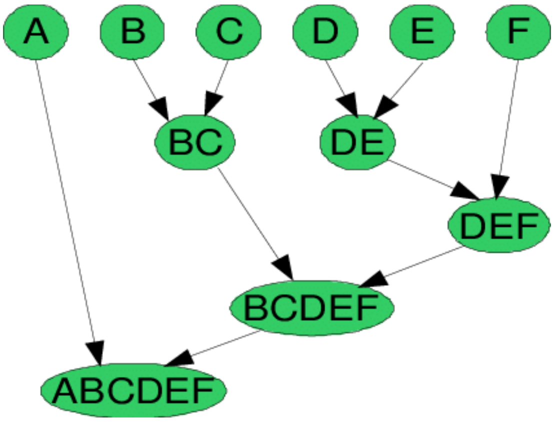
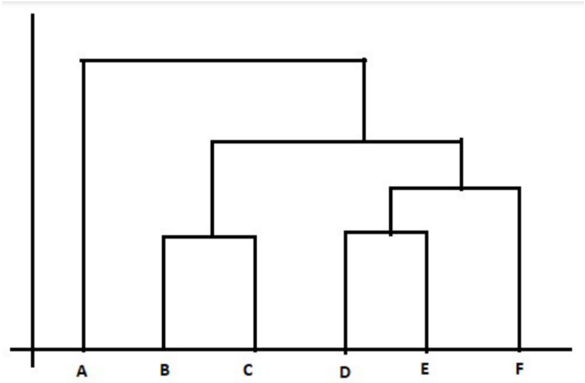
Hierarchical Clustering

Hierarchical clustering is an unsupervised machine learning algorithm that groups unlabelled data points into a single group, named a cluster, and creates a hierarchy of clusters in the form of a "dendrogram". It allows us to see how different sub-clusters relate to each other, and how far apart data points are. (Imagine the structure of files and folders on your computer)

Agglomerative Hierarchical Clustering: If we start with each data point as individual "clusters", we can compare their similarities (compute the proximity matrix) and combine the two closest clusters. After updating your proximity matrix, you can again combine the two closest clusters and repeat this process until only a single cluster remains. Divisive hierarchical clustering is the backwards process of this and is not much used in the real world.





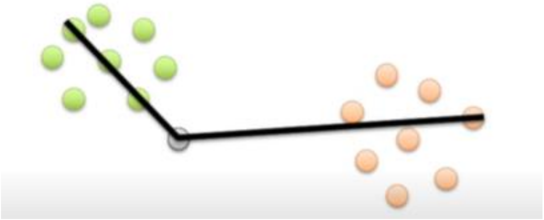
Dendrogram

Y = the Euclidean distances between observations (see below)

X = all the observations present in the given dataset.

Source: https://www.analyticsvidhya.com/blog/2021/06/20-questions-to-test-your-skills-on-hierarchical-clustering-algorithm/

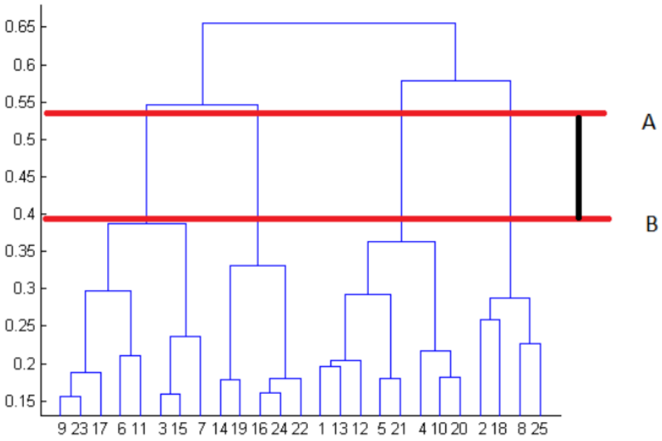
There are different methods for comparing a data point to clusters. You might consider the average of the measurements for each sample, the closest points of each cluster, or the default in R which is the furthest point in each cluster (aka "complete-linkage" - seen below).



Source: https://www.youtube.com/watch?v=7xHsRkOdVwo

How many clusters should we use?

You can set this before you begin. But, as per our requirement according to the problem statement, we can cut the dendrogram at any level. The best choice of number of clusters is the number of vertical lines found within the horizontal intersect that shows the maximum distance vertically without intersecting any other cluster. (above, the optimum is 4 clusters).



Some strengths

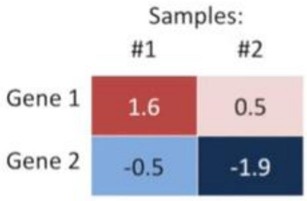
- The model will obtain the optimal number of clusters – humans don't have to do anything.
- Dendograms are easy to understand and offer clear visualisation.
- You can choose a limit for your cluster numbers if you have good domain knowledge.
- You can measure the "goodness" of the clusters in many ways – Dunn's Index is popular.

Some weaknesses

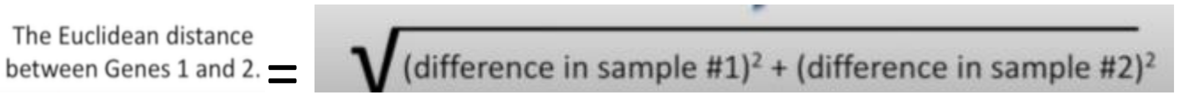
- Space and time complexity (we store the proximity matrix in the RAM, and with each iteration we must update the proximity matrix) mean it is not good for larger datasets.
- Lots of arbitrary decisions, does not work with missing data, does not work well with mixed data types.
- Each of the approaches for calculating similarity between clusters have their own disadvantages (e.g. some cope better with noise than others)

Euclidian distance

The method for determining similarity (proximity) is arbitrarily chosen. However, the Euclidian distance is used a lot:

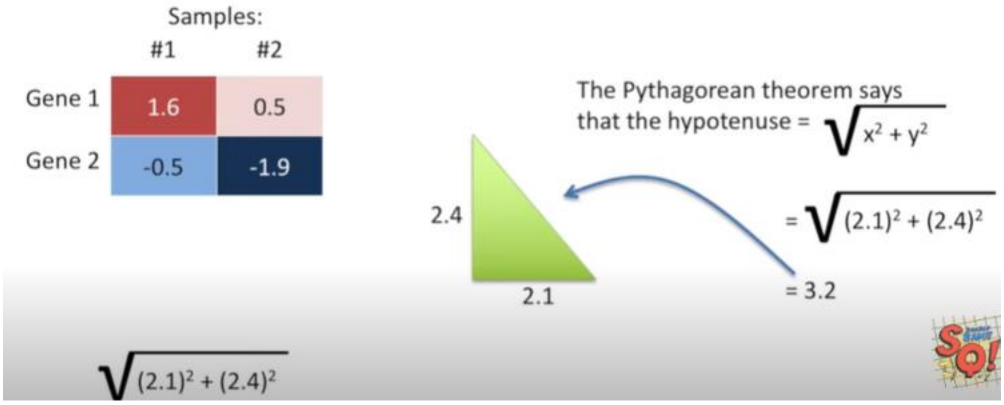


	Samples:	
	#1	#2
Gene 1	1.6	0.5
Gene 2	-0.5	-1.9



$$\text{The Euclidean distance between Genes 1 and 2} = \sqrt{(\text{difference in sample \#1})^2 + (\text{difference in sample \#2})^2}$$

Source: https://www.youtube.com/watch?v=7xHsRkOdVwo



	Samples: #1	#2
Gene 1	1.6	0.5
Gene 2	-0.5	-1.9

The Pythagorean theorem says that the hypotenuse = $\sqrt{x^2 + y^2}$

$= \sqrt{(2.1)^2 + (2.4)^2}$

$= 3.2$

$\sqrt{(2.1)^2 + (2.4)^2}$

The Euclidian distance between Gene 1 and 2 = 3.2

Source: https://www.youtube.com/watch?v=7xHsRkOdVwo

In honour of "Tropical Tuesday" I have created the below hierarchical clustering example (not to scale)