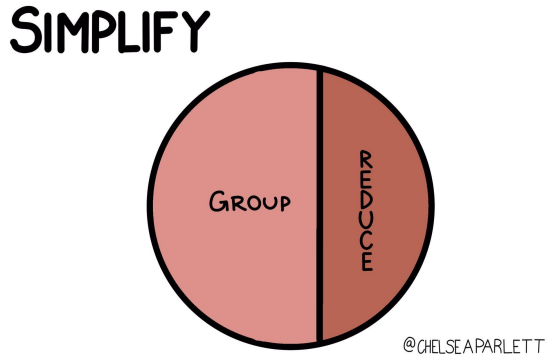
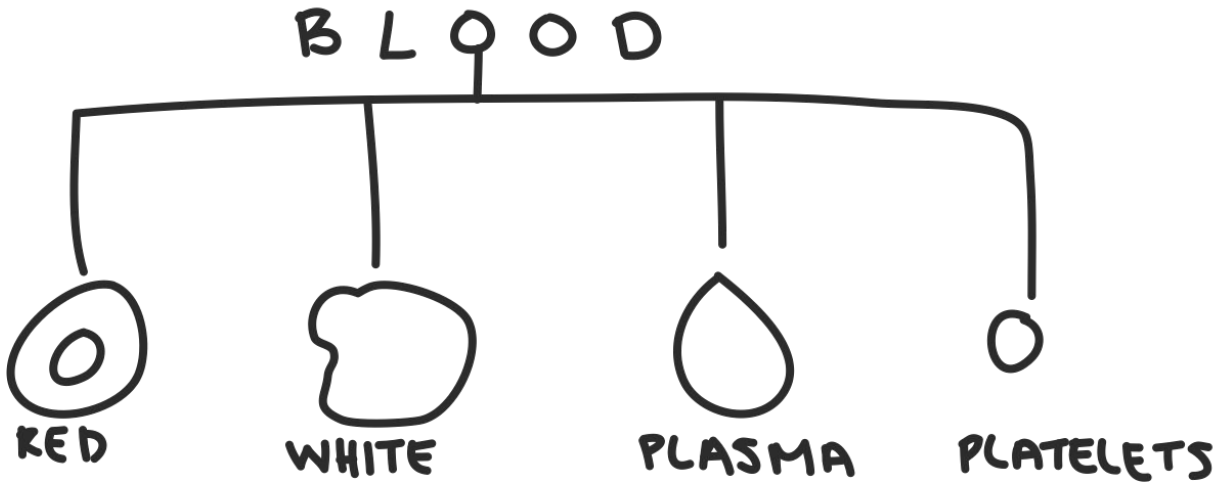


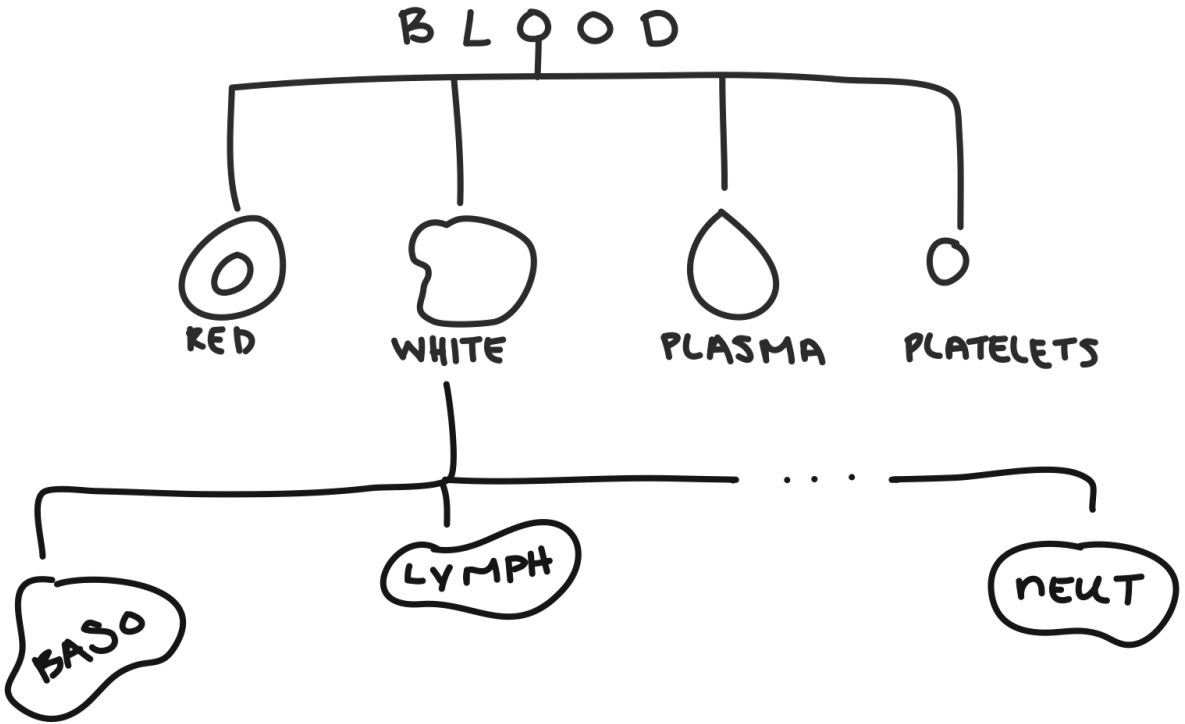
Hierarchical (Agglomerative) Clustering

Dr. Chelsea Parlett-Pelleriti

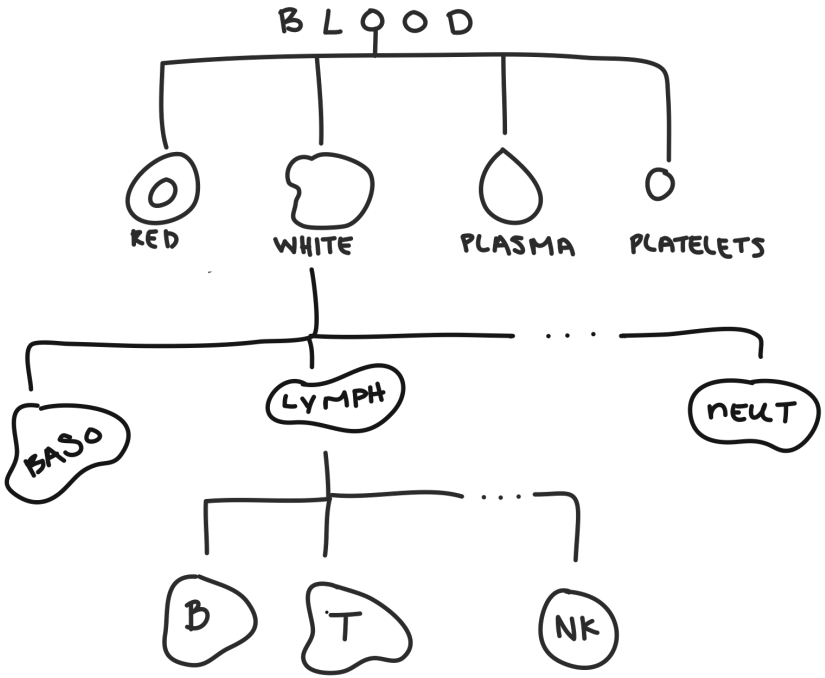
HAC



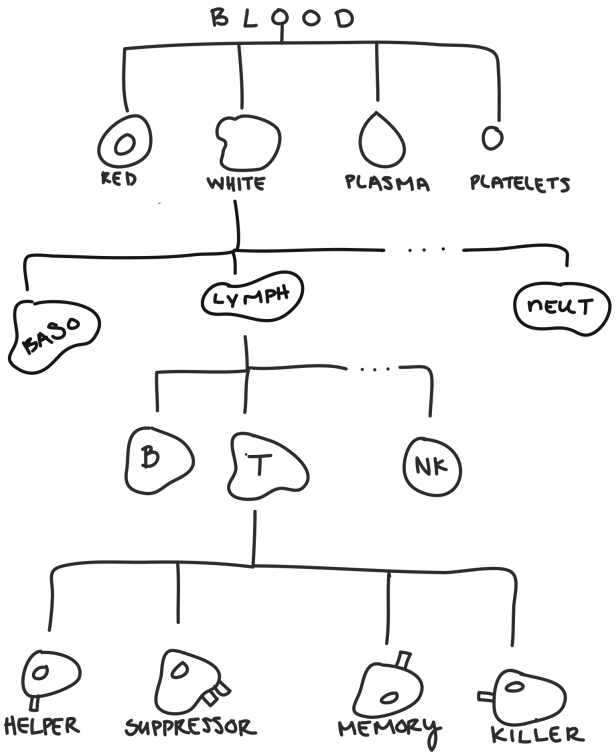
HAC



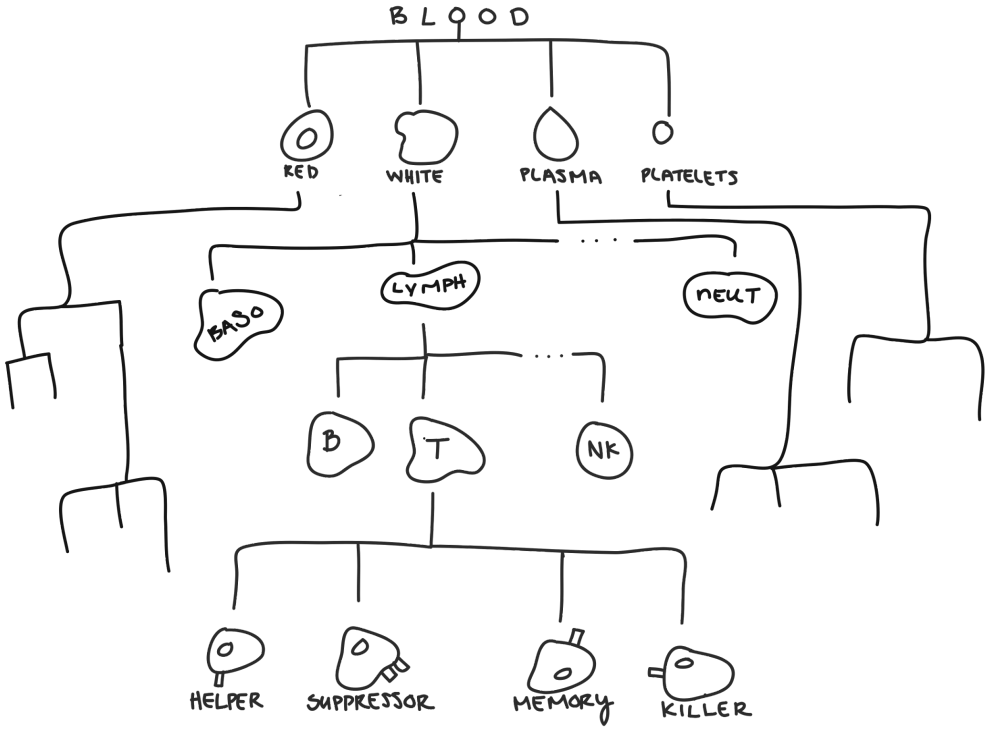
HAC



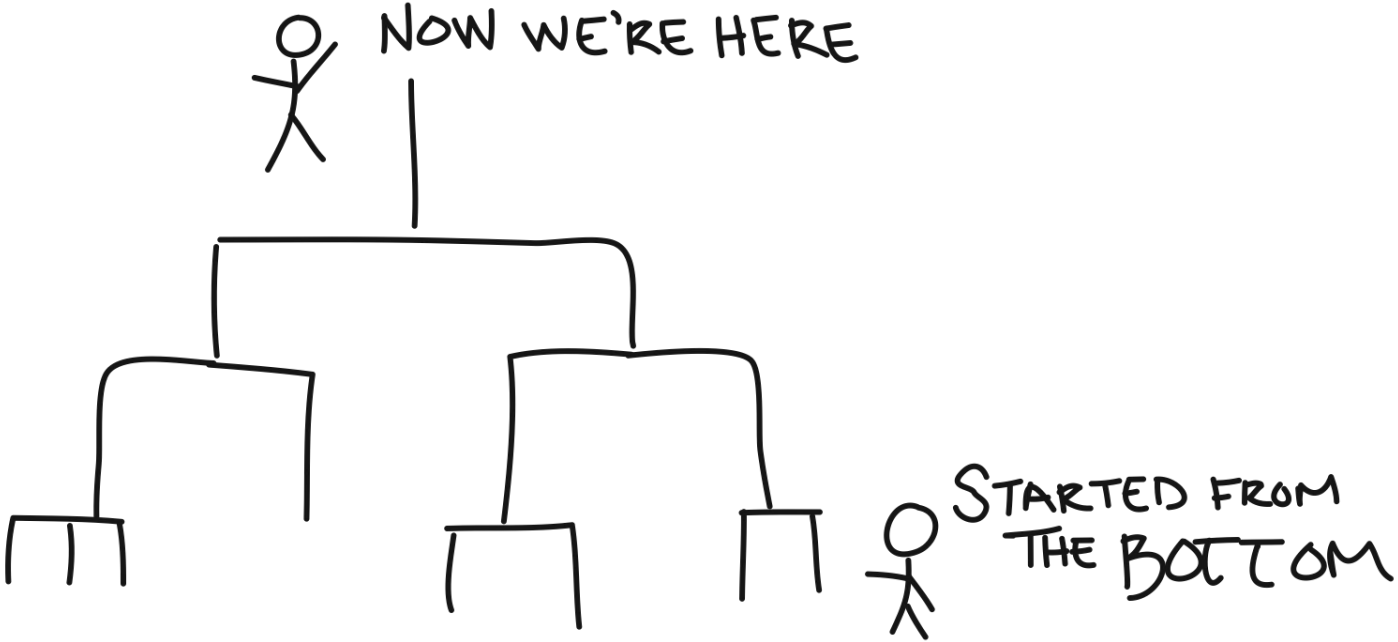
HAC


BLOOD
RED
WHITE
PLASMA
PLATELETS
BASO
LYMPH
NEUT
B
T
NK
HELPER
SUPPRESSOR
MEMORY
KILLER

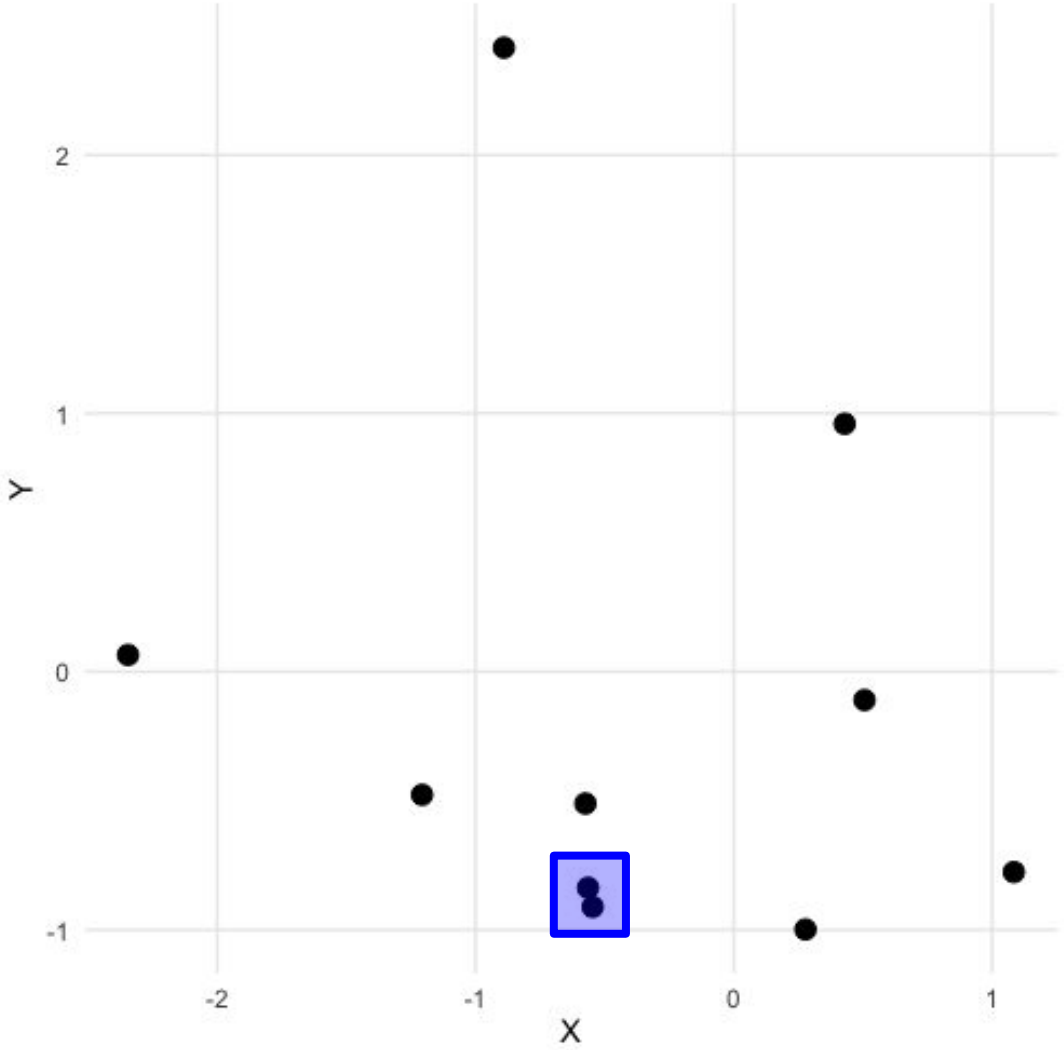
HAC


BLOOD
RED
WHITE
PLASMA
PLATELETS
BASO
LYMPH
NEUT
B
T
NK
HELPER
SUPPRESSOR
MEMORY
KILLER

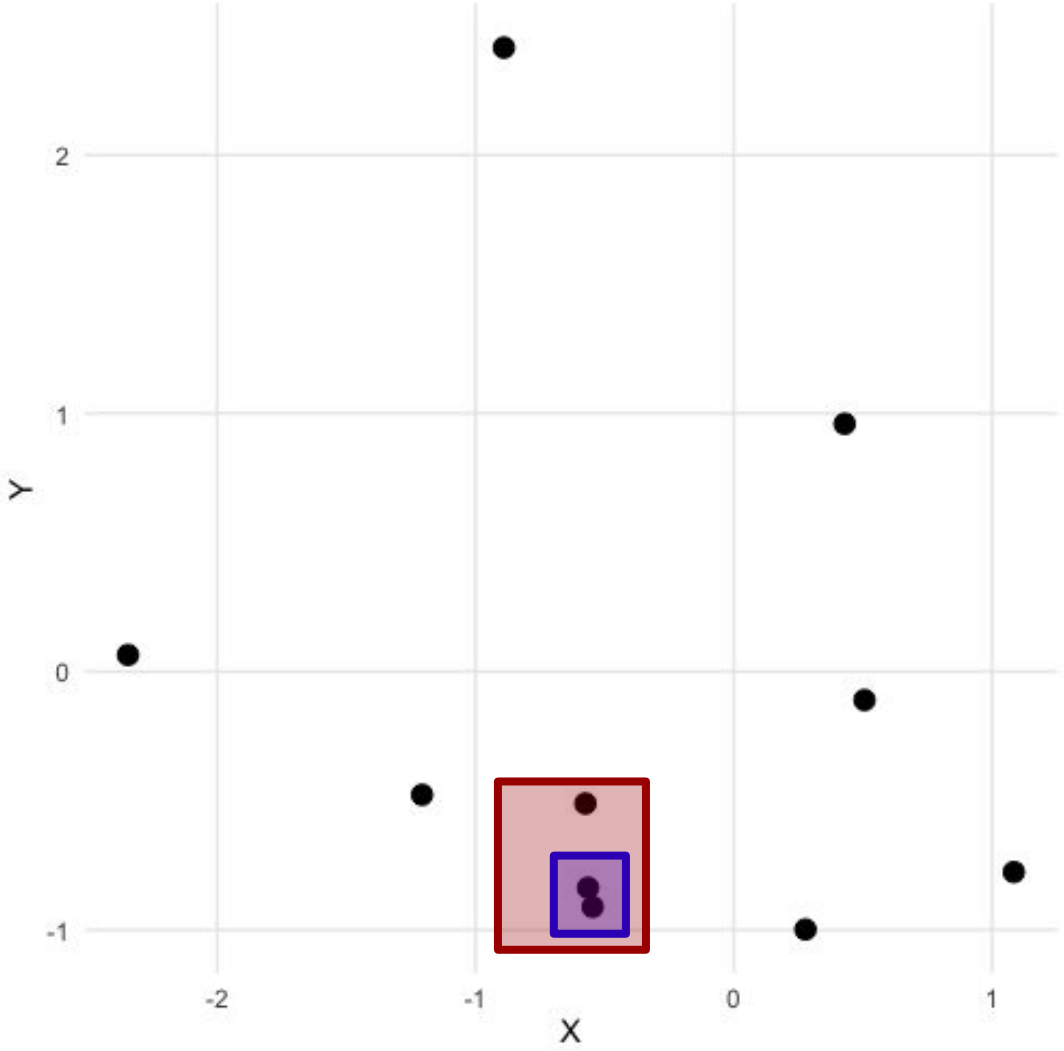
HAC



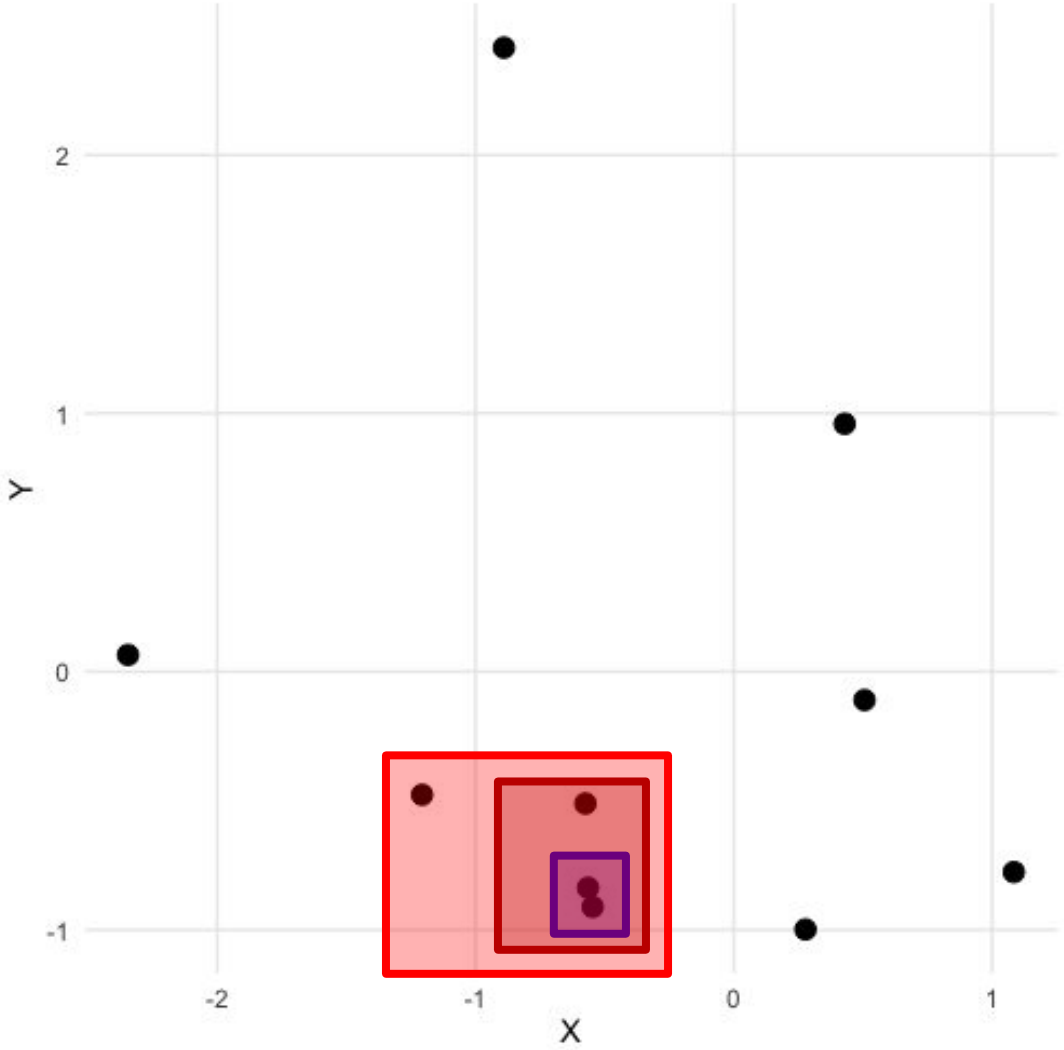
Algorithm



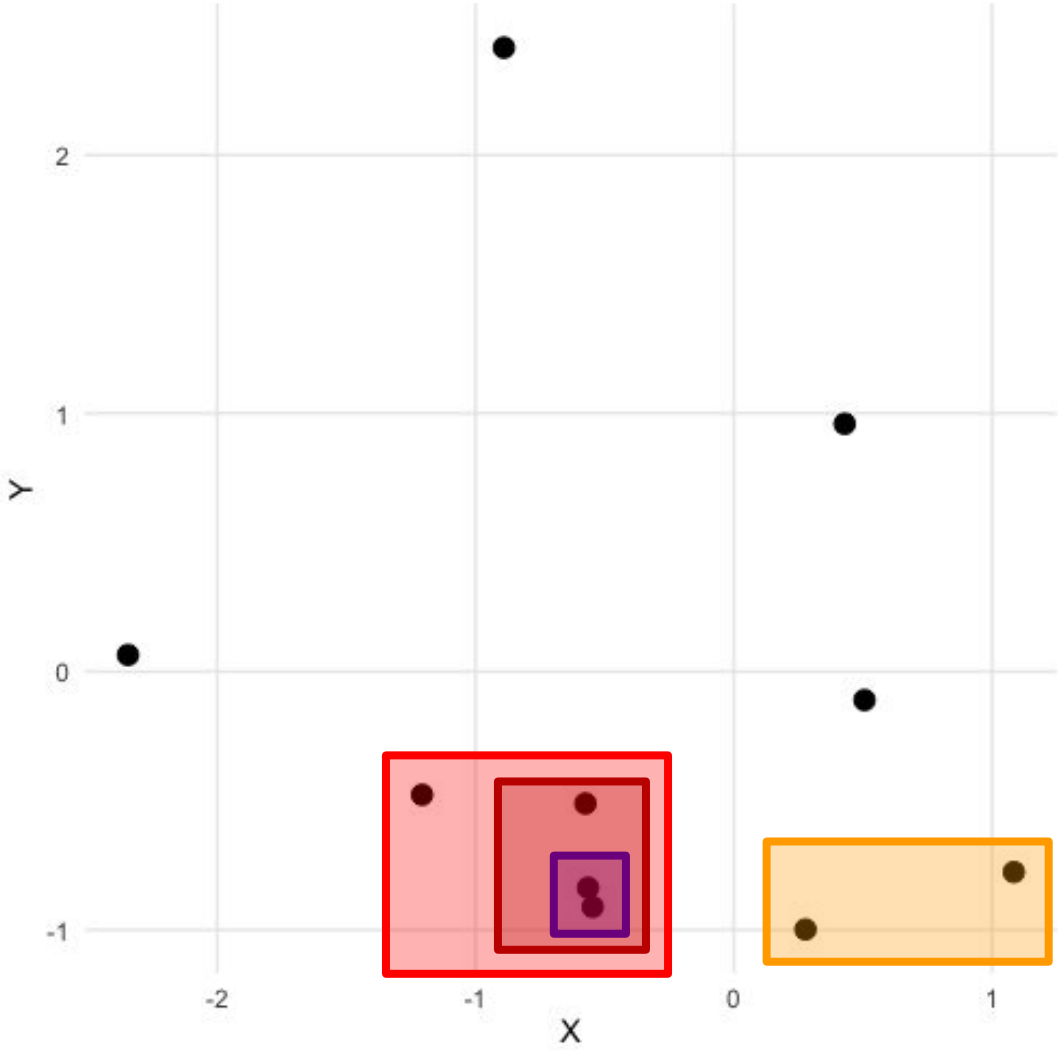
Algorithm



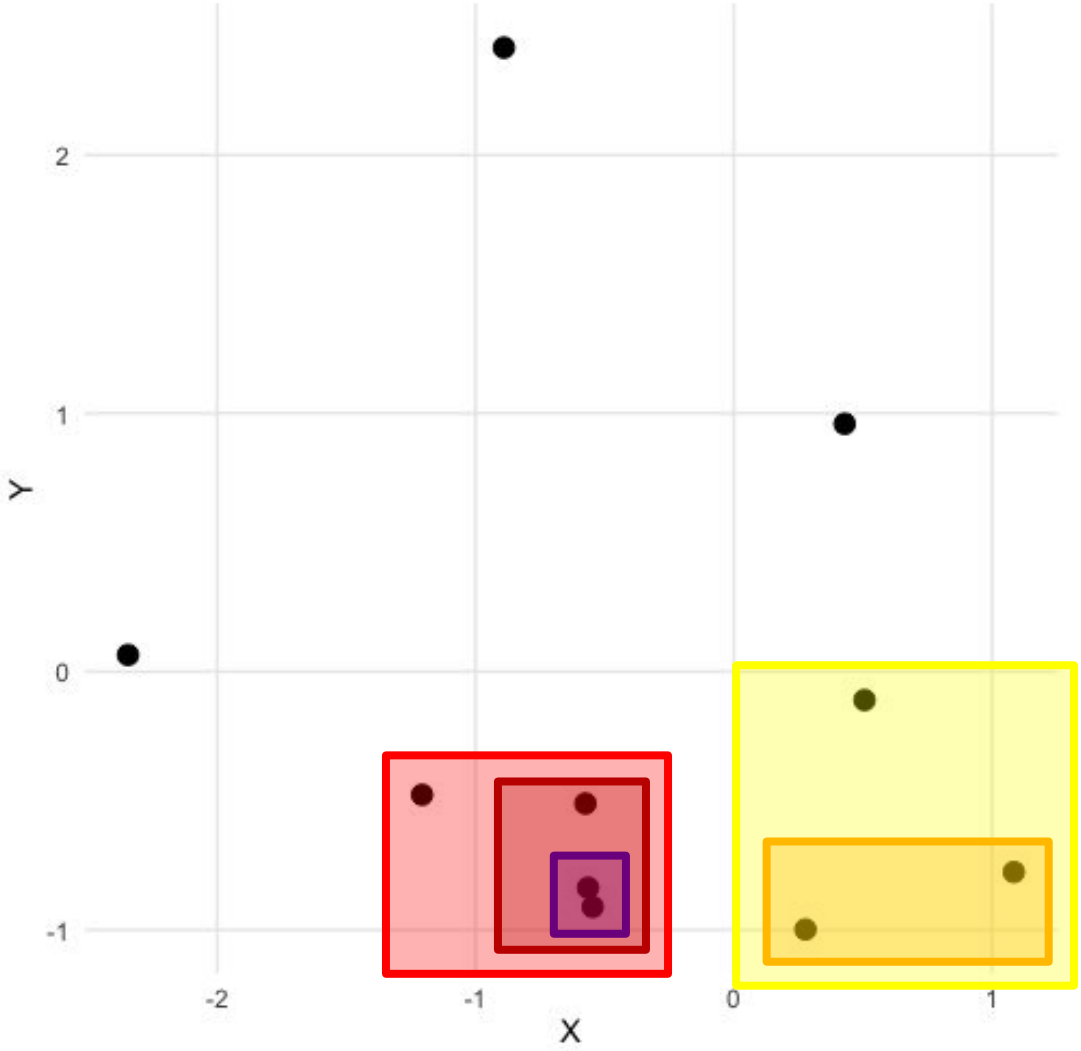
Algorithm



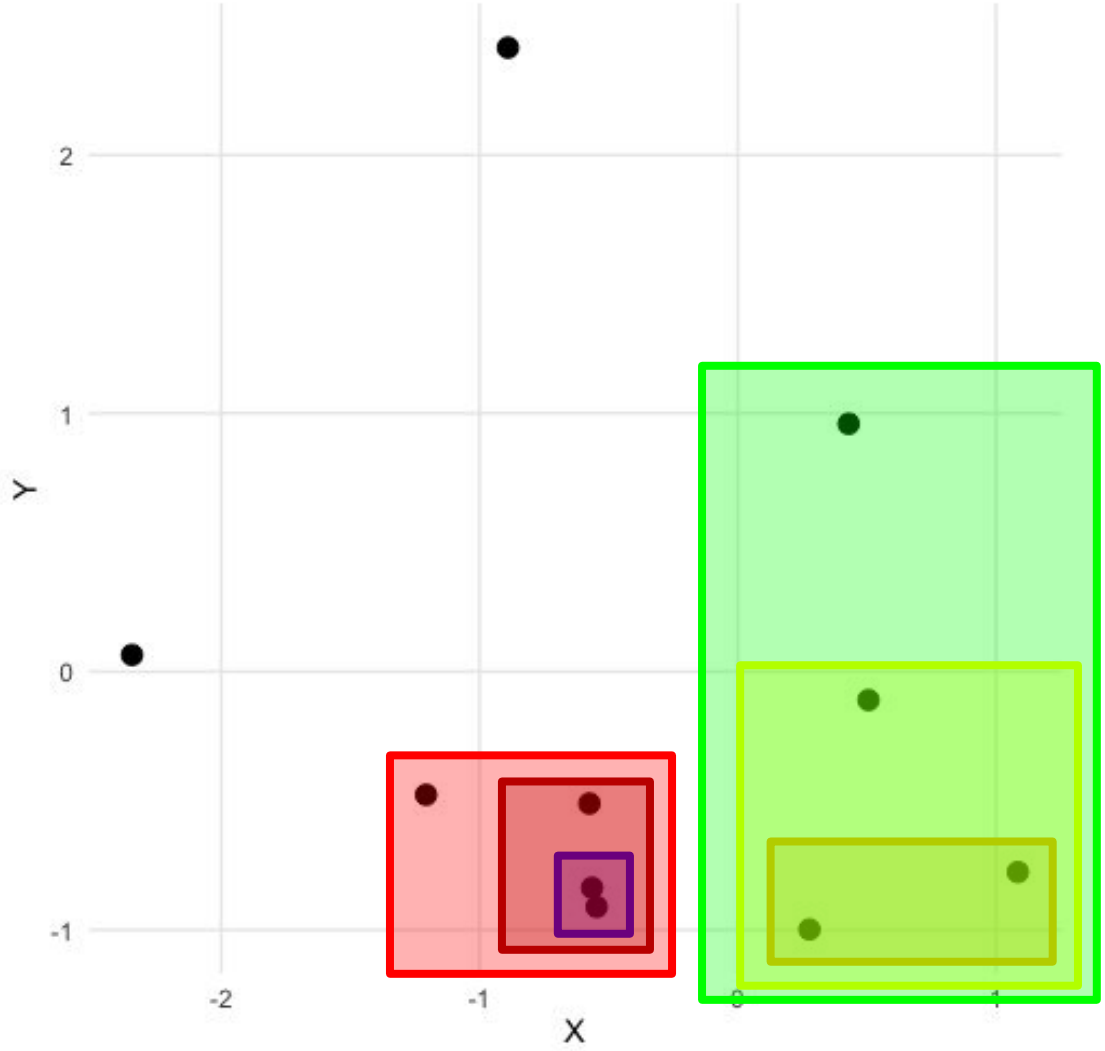
Algorithm



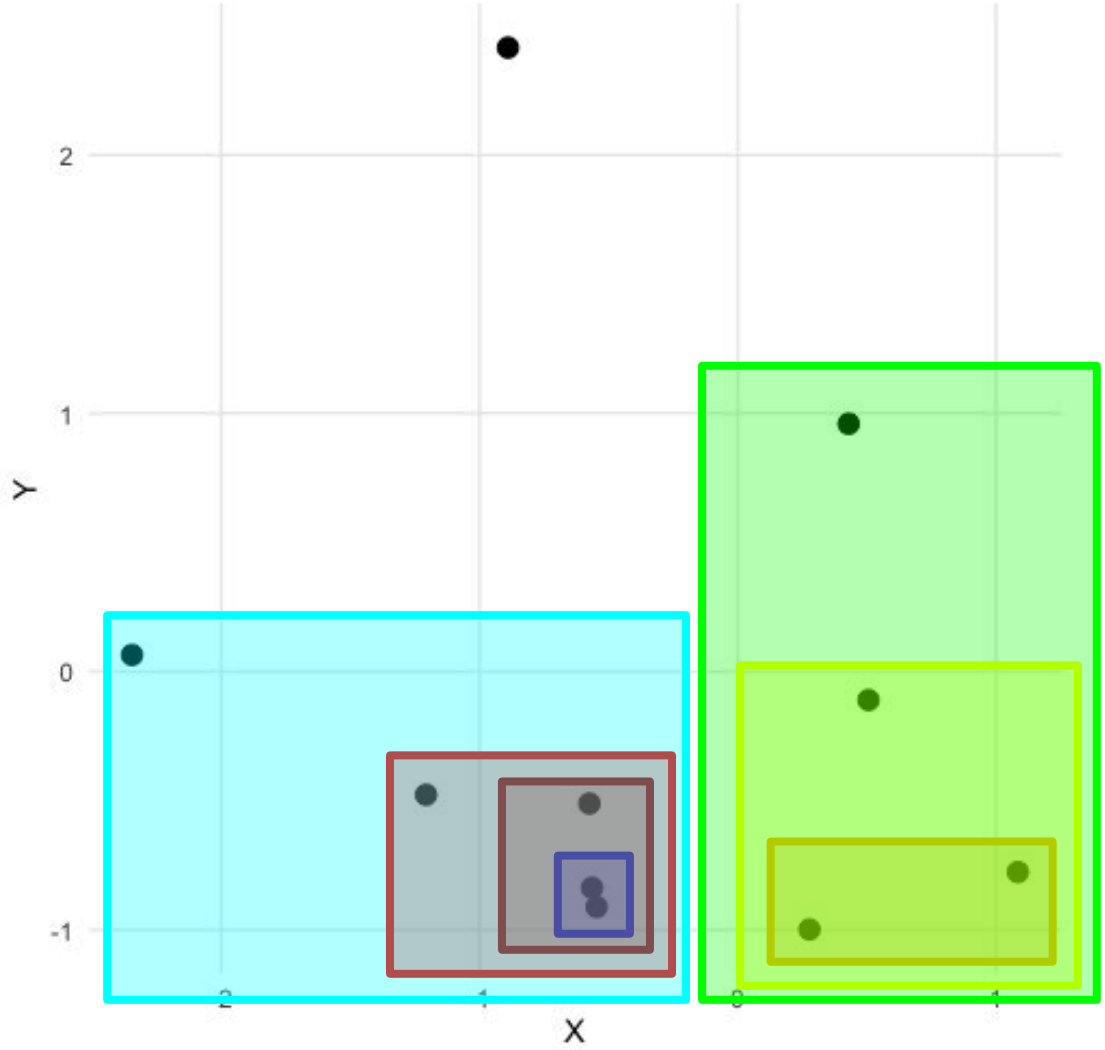
Algorithm



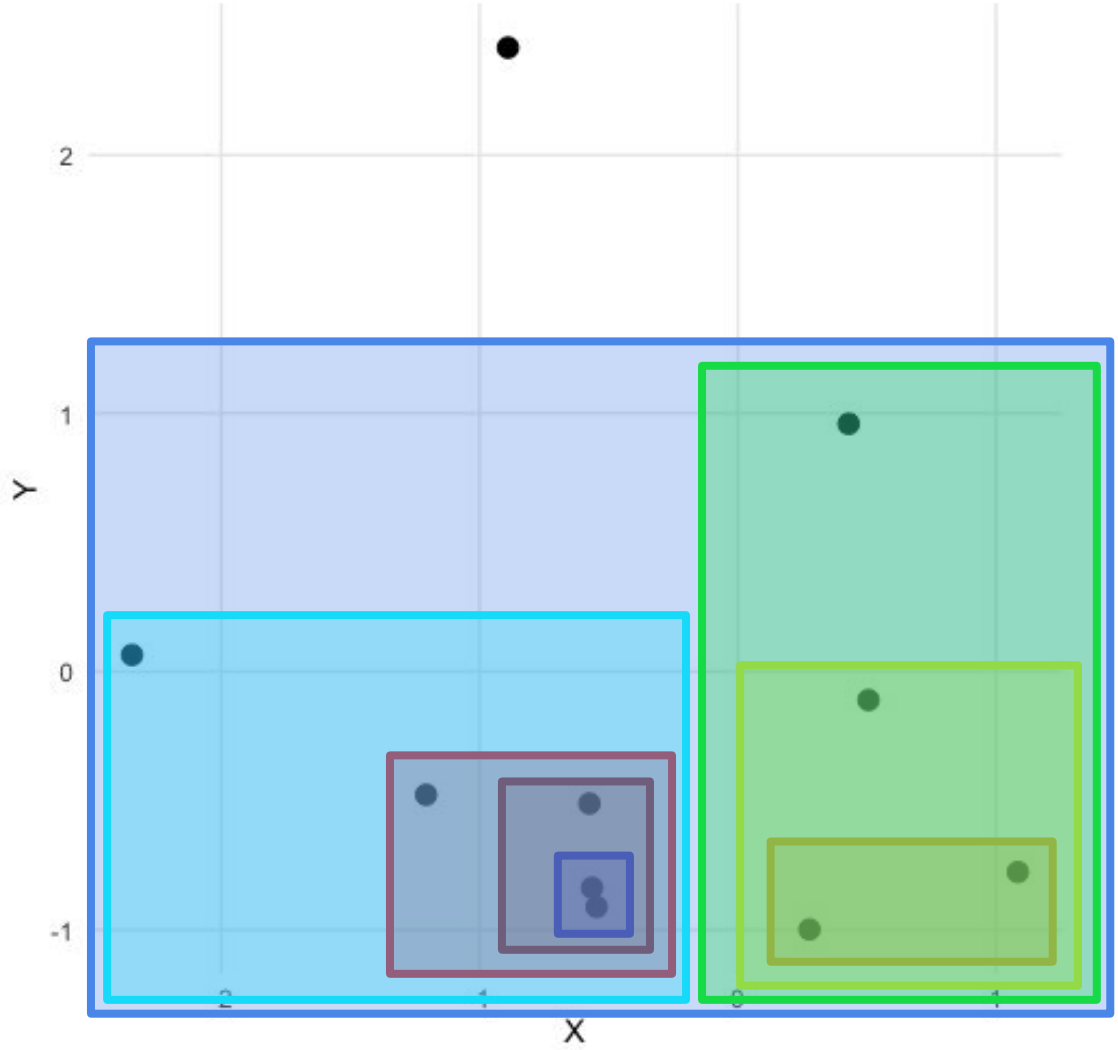
Algorithm



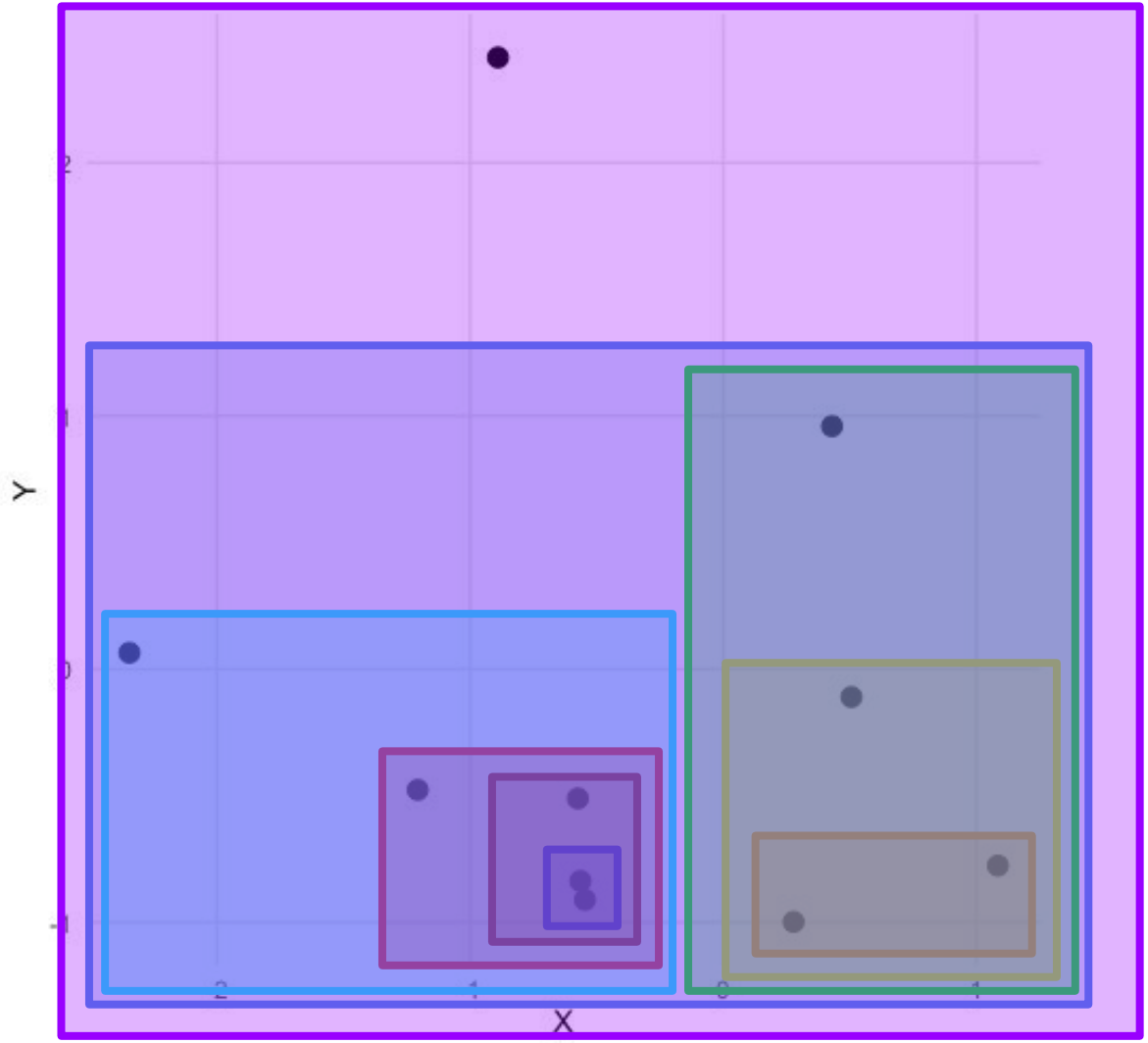
Algorithm



Algorithm



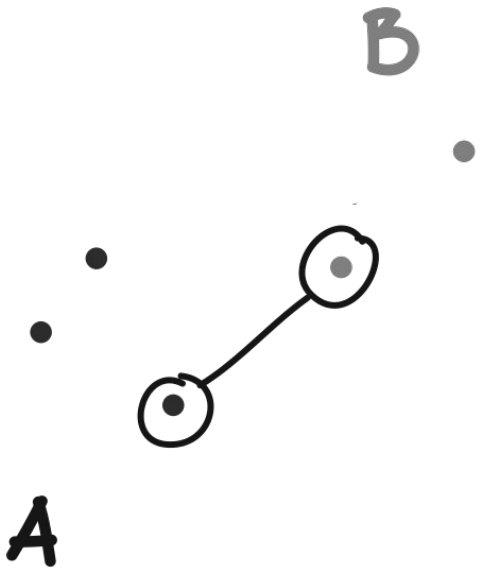
Algorithm



Distance Metrics

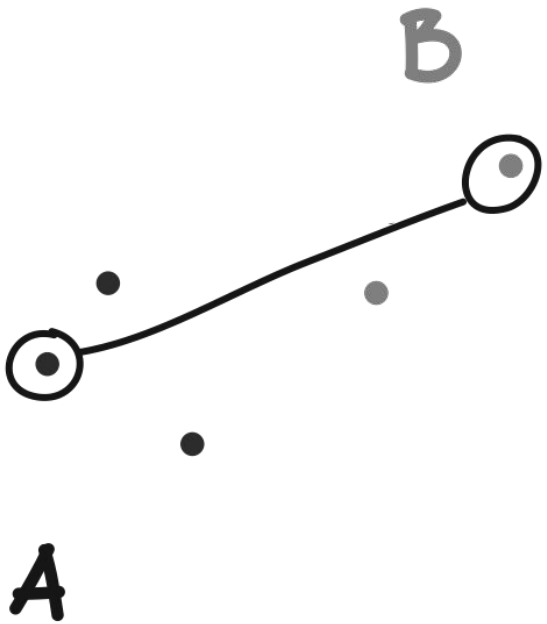
- **Euclidean**: Continuous Data
- **Manhattan**: High Dimensions
- **Hamming**: Categories
- **Cosine**: Word Counts

Linkage Criteria



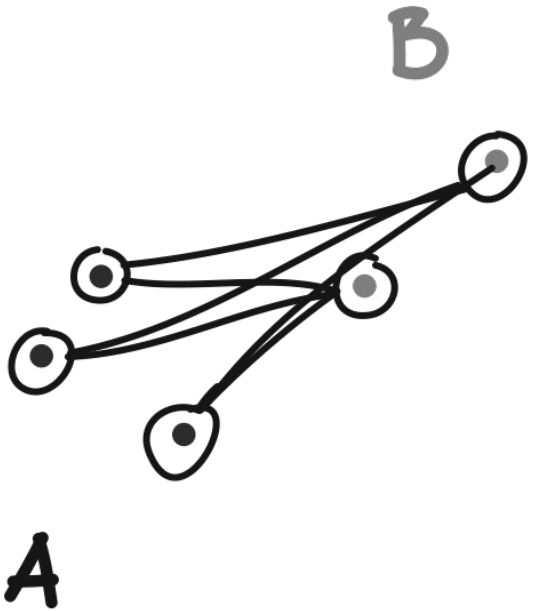
Single

Linkage Criteria



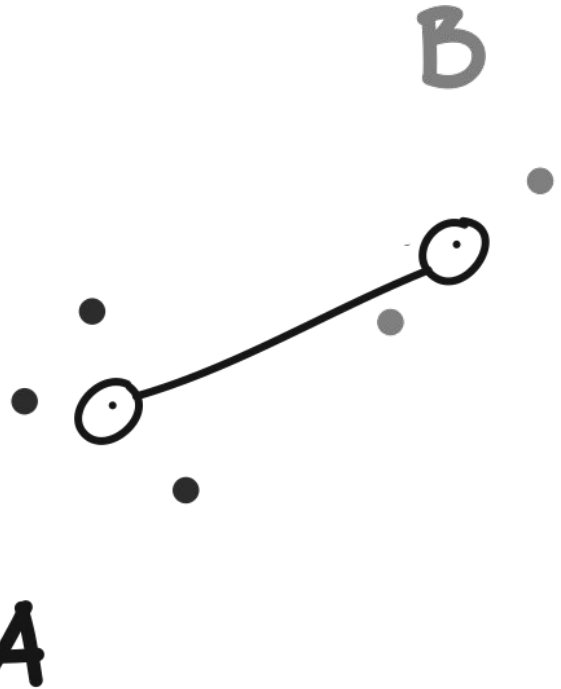
Complete

Linkage Criteria



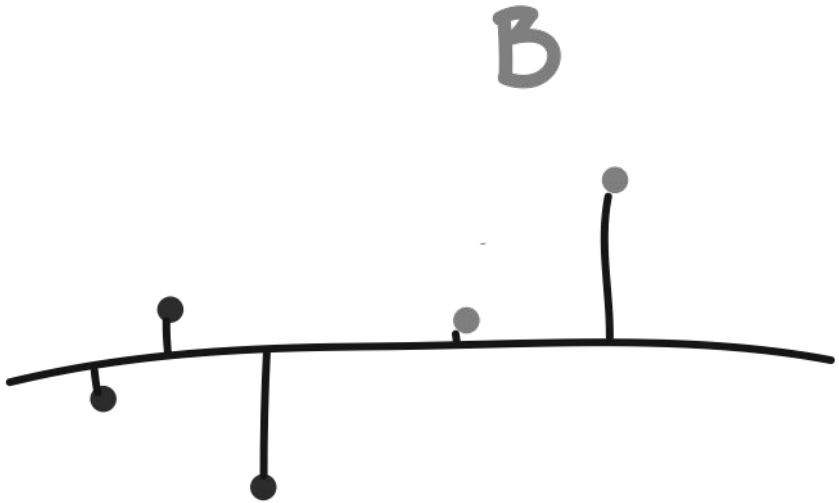
Average

Linkage Criteria



Centroid

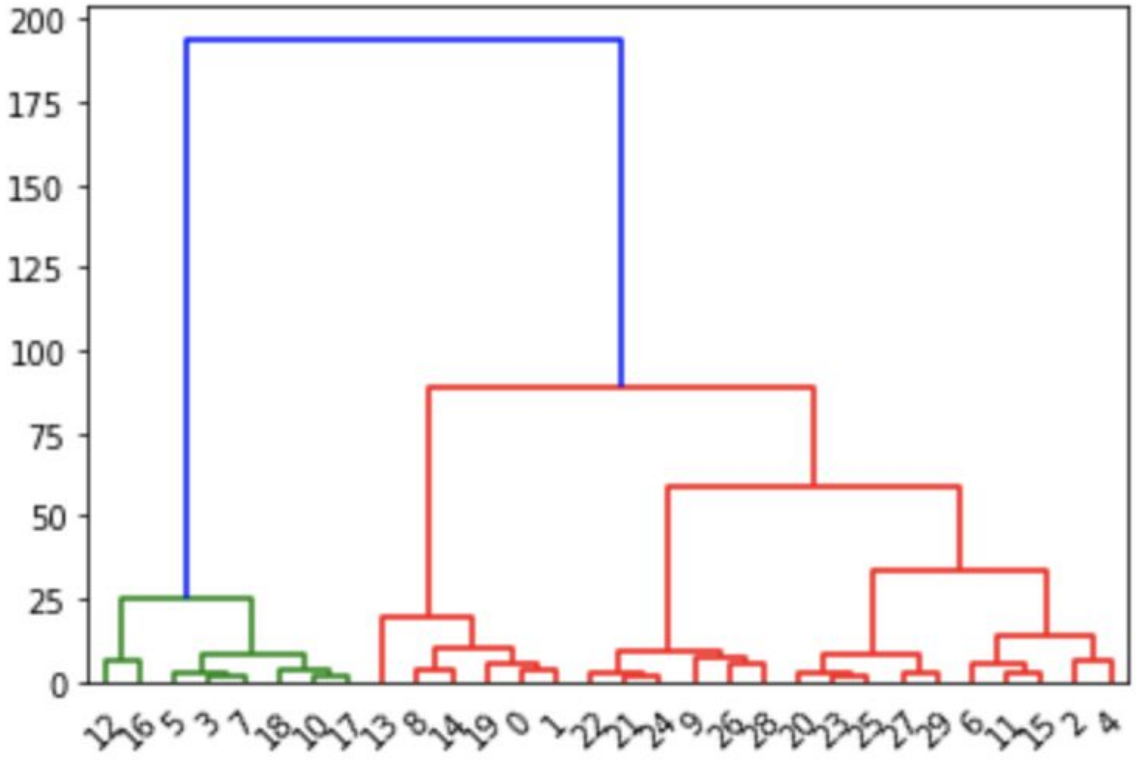
Linkage Criteria



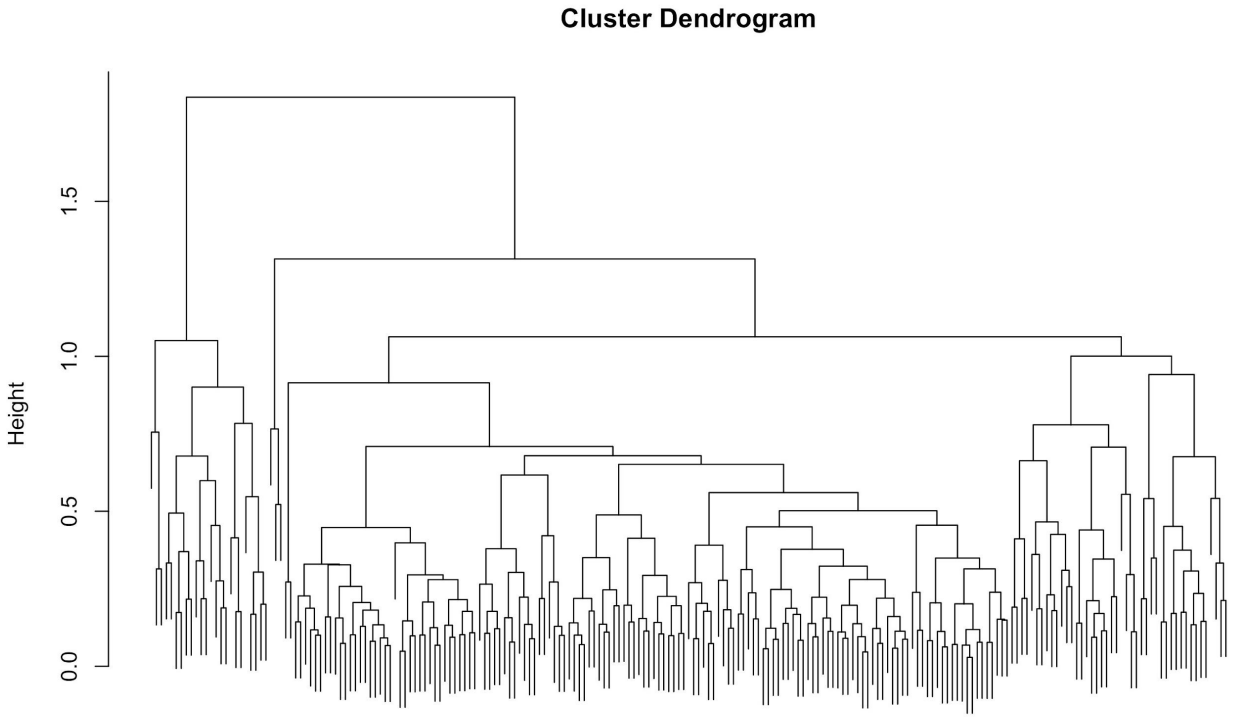
Ward's

A

Reading a Dendrogram



Reading a Dendrogram



HAC

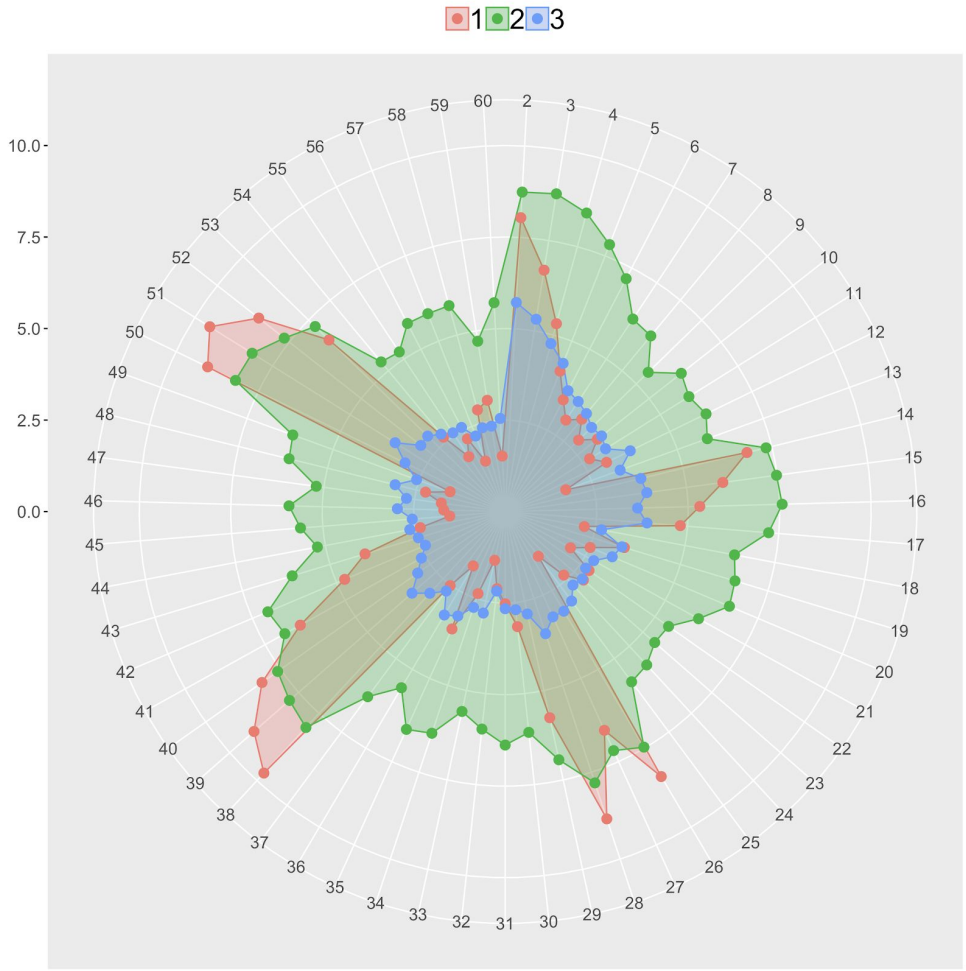
Pros

- Flexible # of clusters
- Model relationship between clusters (hierarchy)
- Flexibility with linkage

Cons

- Very Slow O(n^3)
- Cannot un-merge clusters

My Master’s Thesis



Application

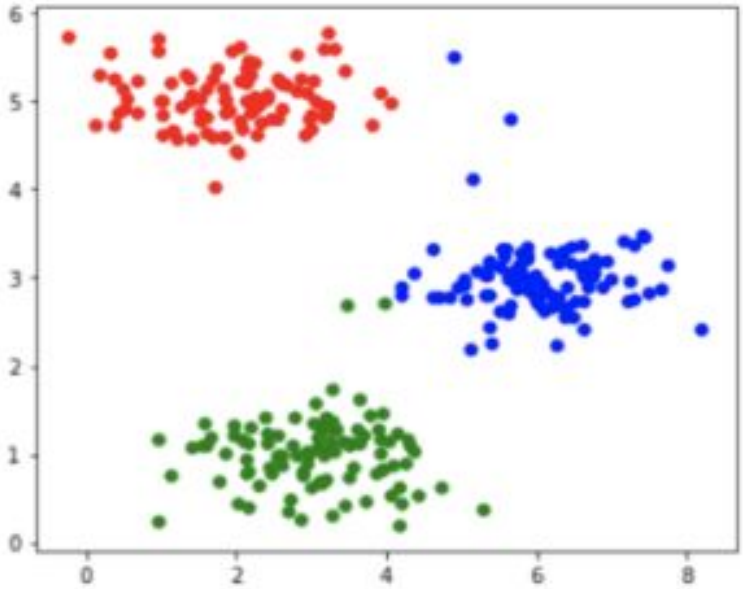


Fig 5: Visualization of Classification.

https://www.researchgate.net/profile/Mohan-Kumar-Av/publication/343878502_Sentiment_Analysis_Using_Robust_Hierarchical_Clustering_Algorithm_for_Opinion_Mining_On_Movie_Reviews-Based_Applications/links/5fe34cea92851c13feb1fe7b/Sentiment-Analysis-Using-Robust-Hierarchical-Clustering-Algorithm-for-Opinion-Mining-On-Movie-Reviews-Based-Applications.pdf

Application

Fig. 3: Grouping results. The samples in the 1^{st} & 2^{nd} lines are from HMDB51 and UCF101 respectively.

https://ieeexplore.ieee.org/document/7423818

Application

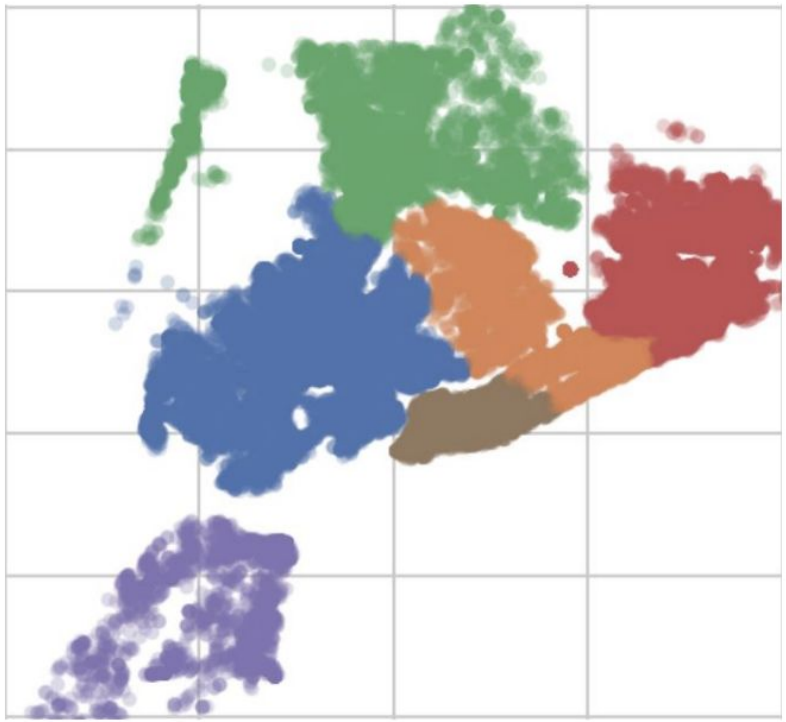


Fig. 5. Clusters computed by HAC on 55k Data Points

https://ieeexplore.ieee.org/abstract/document/9074125

Complexity [edit]

The standard algorithm for **hierarchical agglomerative clustering** (HAC) has a time complexity of $\mathcal{O}(n^3)$ and requires $\Omega(n^2)$ memory, which makes it too slow for even medium data sets. However, for some special cases, optimal efficient agglomerative methods (of complexity $\mathcal{O}(n^2)$) are known: **SLINK**[2] for single-linkage and CLINK[3] for complete-linkage clustering. With a heap, the runtime of the general case can be reduced to $\mathcal{O}(n^2 \log n)$, an improvement on the aforementioned bound of $\mathcal{O}(n^3)$, at the cost of further increasing the memory requirements. In many cases, the memory overheads of this approach are too large to make it practically usable.

Divisive clustering with an exhaustive search is $\mathcal{O}(2^n)$, but it is common to use faster heuristics to choose splits, such as *k*-means.