

# Homework 2

## Pattern Mining and Social Network Analysis

BOUYSSOU Gatien , de POURTALES Caroline, LAMBA Ankit

27 octobre, 2020

### Contents

<b>Continuous and discrete data sets</b>	<b>3</b>
<b>Cluster validation techniques</b>	<b>4</b>
Clustering Tendency . . . . .	4
Why assessing clustering tendency ? . . . . .	4
Methods . . . . .	4
Statistical methods . . . . .	4
Visually . . . . .	4
Determining the Optimal number of Clusters . . . . .	4
Elbow method . . . . .	5
Average silhouette method . . . . .	5
Gap statistic method . . . . .	5
Statistics . . . . .	5
Internal measures . . . . .	5
Compactness . . . . .	6
Separation . . . . .	6
Connectivity . . . . .	6
Silhouette coefficient . . . . .	6
Dunn index . . . . .	6
External mesures . . . . .	6
<b>Principal Components Analysis</b>	<b>7</b>
Different kinds of PCA . . . . .	7
Standard PCA . . . . .	7
Incremental PCA . . . . .	7
Sparse PCA . . . . .	7
Kernel PCA . . . . .	7
Proportion of variance explained (PVE) . . . . .	7
Deciding how many PCs to use . . . . .	7
Example . . . . .	7
<b>Clustering</b>	<b>9</b>
Partitioning Clustering . . . . .	9
K-means . . . . .	9
Within-cluster variation (squared Euclidean distance) . . . . .	9
K-means algorithm . . . . .	9
Choice of k . . . . .	9
Example . . . . .	9
On R . . . . .	9

On python with scikit-learn . . . . .	10
K-medoids algorithm . . . . .	10
Principle . . . . .	10
PAM algorithm (Partitioning Around Medoids) . . . . .	10
CLARA - Clustering Large Applications . . . . .	11
Overall . . . . .	11
Examples . . . . .	11
Hierarchical clustering . . . . .	12
Dissimilarity function . . . . .	12
Euclidean distance . . . . .	12
Correlation-based distance . . . . .	12
Linkage . . . . .	12
Maximum or complete linkage . . . . .	12
Minimum or single linkage . . . . .	12
Mean or average linkage . . . . .	12
Centroid linkage . . . . .	12
Ward's minimum variance method . . . . .	12
P-value . . . . .	13
Examples . . . . .	13
Mushrooms dataset . . . . .	13
On R . . . . .	13
Healthy and non-healthy tissues samples . . . . .	15
On R . . . . .	15
On python with scikit-learn . . . . .	15
Corona dataset . . . . .	16
Fuzzy Clustering . . . . .	16
Overall . . . . .	16
Examples . . . . .	16
<b>Choosing the best algorithm ?</b>	<b>18</b>

## Continuous and discrete data sets

Continuous data is data that can take any value while discrete data can take only certain values. with continuous (distance/similarity based) : Silhouette, Dunn, ...

TO D DISCRETE DATA SET

with discrete (binary, graph based) : modularity, C measure, ...

# Cluster validation techniques

## Clustering Tendency

### Why assessing clustering tendency ?

Clustering can create clusters even if there are no meaningful cluster.

Clustering tendency assessment methods are used to evaluate the validity of clustering analysis. It means evaluate if there are meaningful clusters in the data.

### Methods

**Statistical methods** A method called Hopkins statistic is used to assess the clustering tendency in a data set. It measures the probability for the data set to be generated by an uniform data distribution, it means that this statistic looks at the spatial randomness of the data.

For D, a data set :

Sample n point from D ( $p_1, \dots, p_n$ )

For all  $p_i \in D$ , compute the distance to the nearest neighbor  $x_i < /li > < li > Generate(random\_D)fromarandomuniformdistribution$  ( $q_n$ ) with the same variation as D.

For all  $q_i \in random_D$ , compute the distance to the nearest neighbor  $y_i$

Calculate the Hopkins statistic (H). It is the mean of the nearest neighbor distances in  $random_D$  divided by the sum of the mean nearest neighbor distances in D and  $random_D$ .

$$H = \frac{\sum_i^n y_i}{\sum_i^n X_i + \sum_i^n y_i}$$

If H is about 0.5, it means that the sum along D and  $random_D$  are very close so that the data D is uniformly distributed. It is the null hypothesis, meaning that the data set D is uniformly distributed (no meaningful clusters)

Otherwise we have the Alternative hypothesis: the data set D is not uniformly distributed (contains meaningful clusters).

**Visually** There is an algorithm of the visual assessment of cluster tendency (VAT) approach (Bezdek and Hathaway, 2002).

Compute the dissimilarity matrix using the Euclidean distance measure

Reorder the DM so that similar objects are close to one another. It is now called the ODM.

Display the ODM as an ordered dissimilarity image (ODI) (with some libraries).

Computing the visual form gives us colored squares along the diagonal. The VAT detects the clustering tendency in a visual form by counting them.

## Determining the Optimal number of Clusters

Determining the optimal number of clusters in a data set is fundamental because, for example, in partitioning clustering, such as k-means clustering, it requires the user to specify the number of clusters k.

## Elbow method

The Elbow method looks at the total within-cluster sum of square as a function of the number of clusters.

Compute the clustering algorithm chosen with k going from 1 to 10 for example.

Calculate the within-cluster sum of square for each k

Plot the curve of WSS according to k

The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

An alternative is the Silhouette method.

## Average silhouette method

Average silhouette method computes the average silhouette of observations for different values of k. Then k is chosen if it maximizes the average silhouette.

Compute the clustering algorithm chosen with k going from 1 to 10 for example.

Calculate the average silhouette (i.e. above, AVG.S) of observations for each k

Plot the curve of AVG.S according to k

The location of the maximum is considered as the appropriate number of clusters.

## Gap statistic method

The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference. We choose k when it maximizes the gap statistic so that the clustering structure is far away from the random uniform distribution of points.

Compute the clustering algorithm chosen with k going from 1 to 10 for example.

Calculate the within-cluster sum of square for each k  $W_k$

Generate B reference data sets with a random uniform distribution.

Compute the clustering algorithm chosen on these reference data sets with k going from 1 to  $k_{max}$ .

Calculate the within-cluster sum of square for each k  $W_{kb}$

Compute the gap :

$$Gap(k) = \frac{1}{B} \sum_{b=1}^B \log(W_{kb}) - \log(W_k)$$

Compute the standard deviation of the statistics  $s_k$ .

Choose the smallest k such that the gap statistic is within one standard deviation of the gap at k+1.

$$Gap(k+1) > Gap(k) - s_{k+1}$$

## Statistics

### Internal measures

Internal cluster validation uses the internal information of the clustering process to evaluate the clustering structure without any external information.

Internal validation measures reflect often the compactness, the connectedness and the separation of the cluster partitions.

**Compactness** Compactness measures how close are the objects within the same cluster. We evaluate it with the notion of distance such as the cluster-wise within average/median distances between observations.

**Separation** Separation measures how well clusters are separated from one another. We evaluate it by looking at the distances between clusters' centers or with the pairwise minimum distances between objects in different clusters.

**Connectivity** Connectivity corresponds to what extent items are placed in the same cluster as their nearest neighbors in the data space.

**Silhouette coefficient** The Silhouette coefficient evaluates the performance of your clustering model on a dataset. It measures how well an observation is clustered and it estimates the average distance between clusters.

This coefficient can take 3 value :

- 1 it means that the cluster is far away from its neighbours
- 0 indicates that it is close from one or multiple clusters
- -1 or negative values in general means that the cluster is allocated to the wrong values

It is possible to compute this coefficient thanks to the following formula :

$$Silhouette_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Where :

- $a_i$  is the average distance between the observation i within its cluster
- $b_i$  is the average distance between the observation i and all the observations belonging to another cluster

How to interpret the value of  $S_i$  ?

A large  $S_i$ , almost 1, means the observation is very well clustered.

A small  $S_i$ , close to 0, means that the observation lies between two clusters.

A negative  $S_i$  means the observation is probably placed in the wrong cluster.

**Dunn index** The Dunn index is another internal clustering validation measure.

It is the minimal distance between two clusters (the smallest separation) over the maximal distance between the objects of one clusters (the biggest diameter of a cluster).

$$Dunn\ index = \frac{min.separation}{max.diameter}$$

Compact and well-separated clusters in a data set means a small diameter of the clusters and a large distance between the clusters. Thus, Dunn index should be maximized.

## External mesures

External cluster validation compares the results of a cluster analysis with an externally known result. It measures how well a cluster match extern class labels. We know k, the number of clusters, in advance. So we use this validation to choose the correct clustering method.

We compare the identified clusters to an external reference.

# Principal Components Analysis

The goal of PCA is to identify which features in the dataset explain the most variability.

TO-DO

## Different kinds of PCA

### Standard PCA

TO-DO

### Incremental PCA

TO-DO

### Sparse PCA

TO-DO

### Kernel PCA

TO-DO

## Proportion of variance explained (PVE)

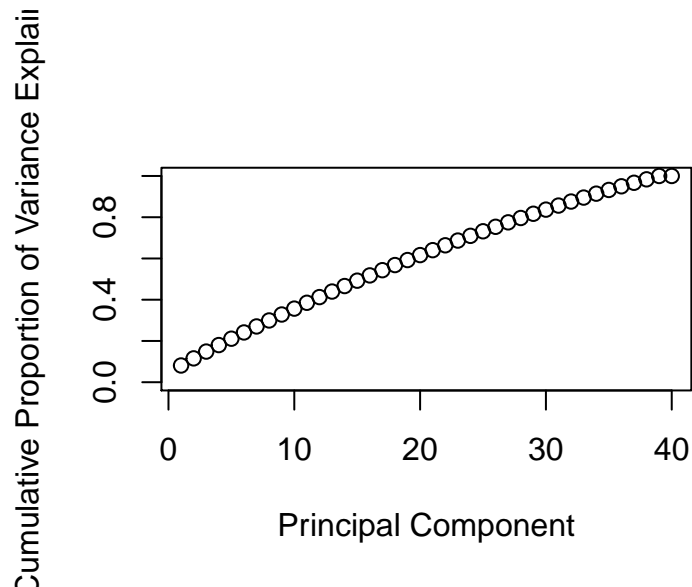
TO-DO

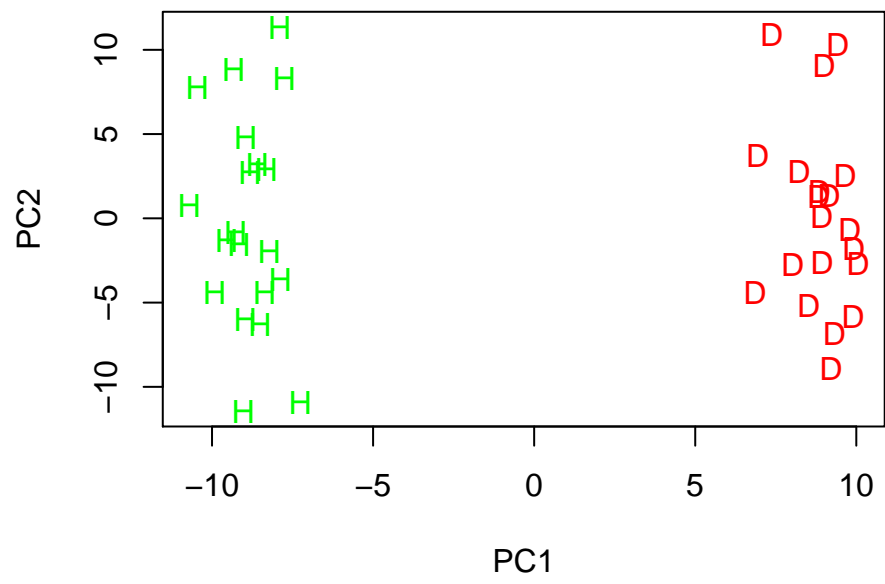
## Deciding how many PCs to use

TO-DO

## Example

The following dataset consists of 40 tissue samples with measurements of 1,000 genes. The first 20 tissues come from healthy patients (H) and the remaining 20 come from a diseased patient group (D).







# Clustering

## Partitioning Clustering

### K-means

The objective of clustering is to distinct groups from the datatest. With k-means we want to distinct k groups. The algorithm will assign each observation to exactly one of the cluster. It optimizes the groups by minimizing the within-cluster variation such that the sum of the with-cluster variations across all the clusters is the smallest possible.

**Within-cluster variation (squared Euclidean distance)** If  $\mu_k$  is the center of the cluster k. The total with-cluster variation is TW :

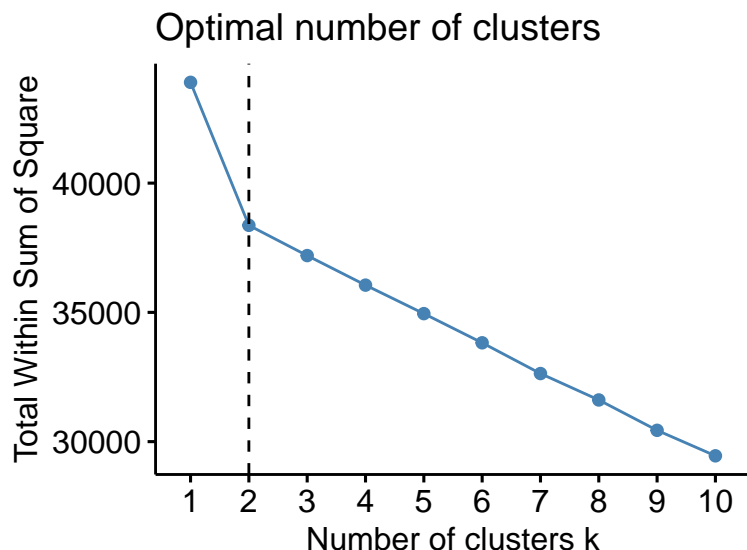
$$TW = \sum_{j=1}^k W_j = \sum_{j=1}^k \sum_{x_i \in C_j} (x_i - \mu_k)^2$$

**K-means algorithm** The first step when using k-means clustering is to indicate the number of clusters (k) that will be generated in the final solution. The algorithm starts by randomly selecting k objects from the data set to serve as the initial centers for the clusters. The selected objects are also known as cluster means or centroids.

**Choice of k** We compute k-means clustering using different k, then we choose the number of cluster according to the location of a bend on the graph representing the Within-cluster variation according to k.

### Example

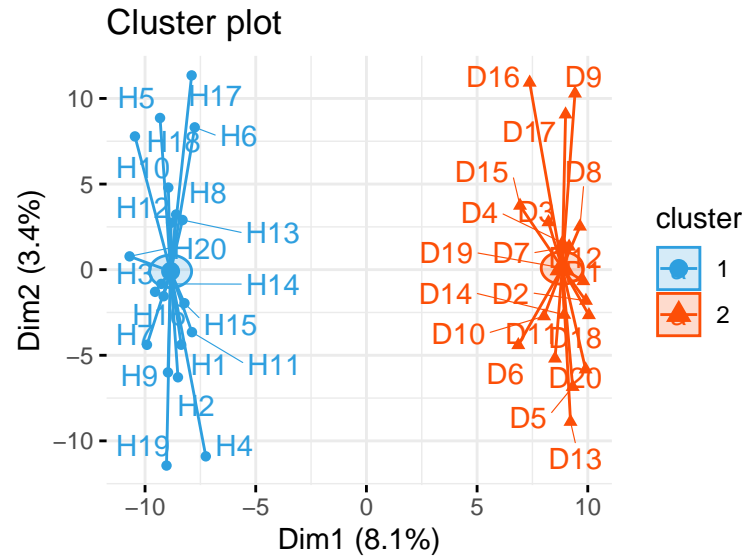
**On R** According to this graph, we should choose k=2 (it makes sense since we have Healthy and non healthy patients).



Then applying kmeans with 2 clusters we observe that the 20 first individuals (healthy) are not in the same cluster than the 20 others (non healthy).

##	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20
##	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
##	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
##	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Since we have a multi-dimensional dataset, we apply dimensionality reduction with the use of PCA to plot the clusters. On the x axis, it is the first PCA, on the y axis, it is the second PCA.



**On python with scikit-learn** By applying Kmeans (with 2 clusters) from scikit-learn on the gene dataset, we have the following assignation to clusters. The 20 first individuals (Healthy) are well separated from the 20 last individuals since there are not in the same cluster.

```
## array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
##        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

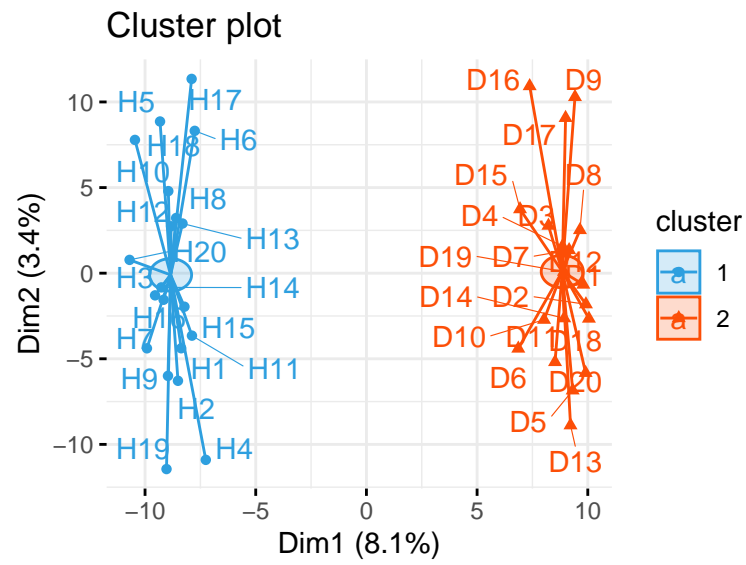
### K-medoids algorithm

**Principle** The k-medoids algorithm is a clustering approach related to k-means clustering. In k-medoids clustering, each cluster is represented by one of the data point in the cluster.

The most common k-medoids clustering methods is the PAM.

### PAM algorithm (Partitioning Around Medoids)

```
## H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19 H20
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```



**CLARA - Clustering Large Applications**

**Overall** TO DO

**Examples** TO DO

## Hierarchical clustering

### Dissimilarity function

#### Euclidean distance

##		H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
## H1		0.0	45.2	44.5	45.6	45.9	45.2	45.0	45.8	45.0	46.0
## H2		45.2	0.0	45.7	44.6	44.4	45.5	44.8	44.6	44.4	44.5
## H3		44.5	45.7	0.0	43.9	46.0	44.9	45.7	45.2	44.1	45.3
## H4		45.6	44.6	43.9	0.0	47.4	45.1	45.6	45.7	44.0	44.4
## H5		45.9	44.4	46.0	47.4	0.0	45.4	44.6	45.6	45.7	44.3
## H6		45.2	45.5	44.9	45.1	45.4	0.0	44.9	43.9	44.7	43.3
## H7		45.0	44.8	45.7	45.6	44.6	44.9	0.0	45.3	43.5	44.4
## H8		45.8	44.6	45.2	45.7	45.6	43.9	45.3	0.0	44.0	44.1
## H9		45.0	44.4	44.1	44.0	45.7	44.7	43.5	44.0	0.0	44.2
## H10		46.0	44.5	45.3	44.4	44.3	43.3	44.4	44.1	44.2	0.0

**Correlation-based distance** Correlation-based distance considers two observations to be similar if their features are highly correlated, even though the observed values may be far apart in terms of Euclidean distance.

##		H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
## H1		0.0	1	1.0	1.0	1.0	1.0	1.0	1.1	1.0	1.1
## H2		1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
## H3		1.0	1	0.0	0.9	1.0	1.0	1.0	1.0	1.0	1.0
## H4		1.0	1	0.9	0.0	1.1	1.0	1.0	1.0	1.0	1.0
## H5		1.0	1	1.0	1.1	0.0	1.0	1.0	1.0	1.0	1.0
## H6		1.0	1	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.9
## H7		1.0	1	1.0	1.0	1.0	1.0	0.0	1.0	0.9	1.0
## H8		1.1	1	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0
## H9		1.0	1	1.0	1.0	1.0	1.0	0.9	1.0	0.0	1.0
## H10		1.1	1	1.0	1.0	1.0	0.9	1.0	1.0	1.0	0.0

#### Linkage

The linkage function takes the distances and groups pairs of objects into clusters based on their similarity. These clusters are then linked to each other to create bigger clusters and the linkage continues until all the data are linked together in a hierarchical tree.

**Maximum or complete linkage** The distance between two clusters is defined as the maximum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce more compact clusters.

**Minimum or single linkage** The distance between two clusters is defined as the minimum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce long, “loose” clusters.

**Mean or average linkage** The distance between two clusters is defined as the average distance between the elements in cluster 1 and the elements in cluster 2.

**Centroid linkage** The distance between two clusters is defined as the distance between the centroid for cluster 1 (a mean vector of length  $p$  variables) and the centroid for cluster 2.

**Ward’s minimum variance method** It minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.

## P-value

Generate bootstraps samples

Compute hierarchical clustering on each bootstrap copy

Compute for each cluster :

The bootstrap probability (BP) value : frequency that the cluster is identified in bootstrap copies.

The approximately unbiased (AU) probability values (p-values) by multiscale bootstrap resampling. Clusters with p-value above 95% are considered to be strongly supported by data.

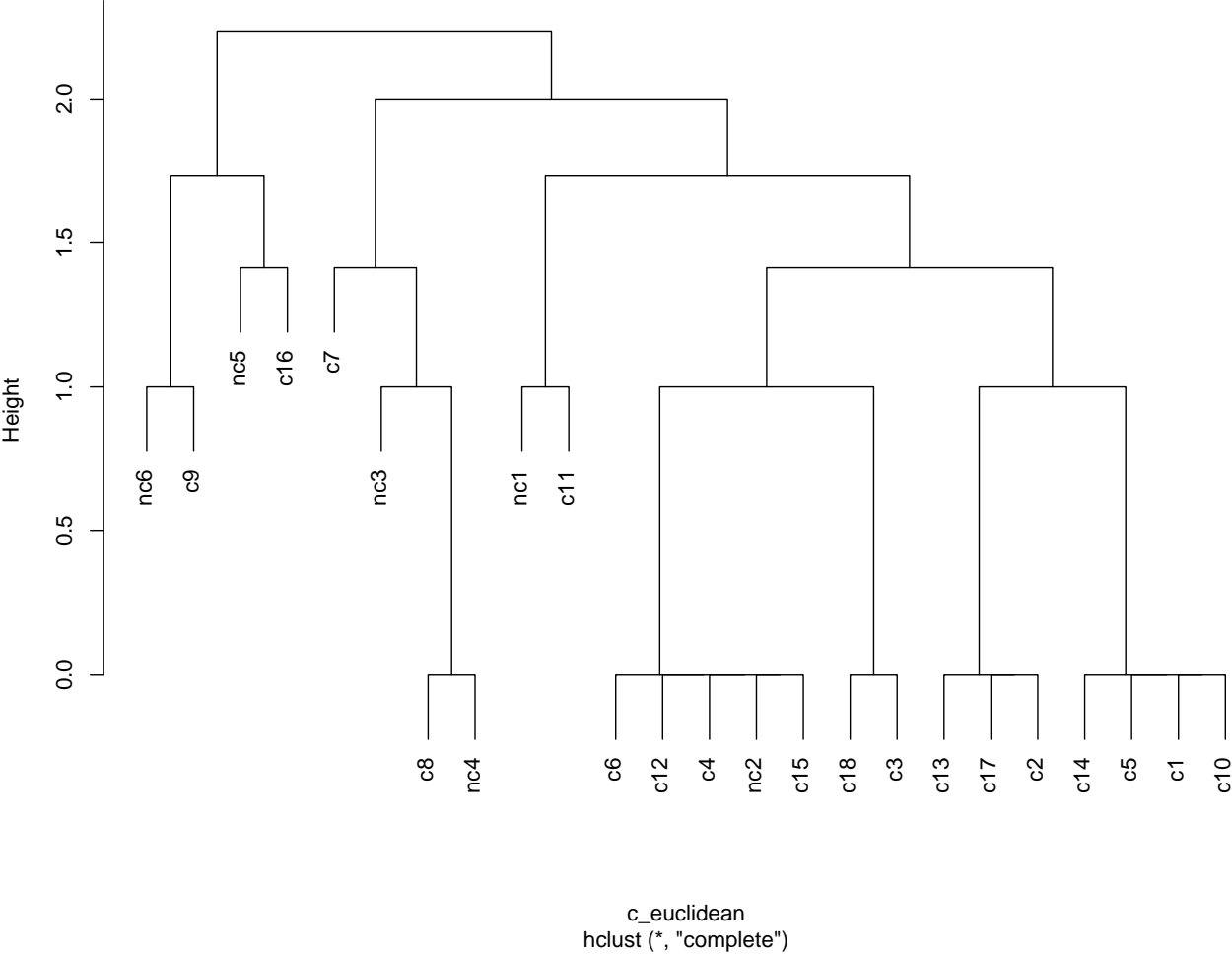
## Examples

### Mushrooms dataset

#### On R

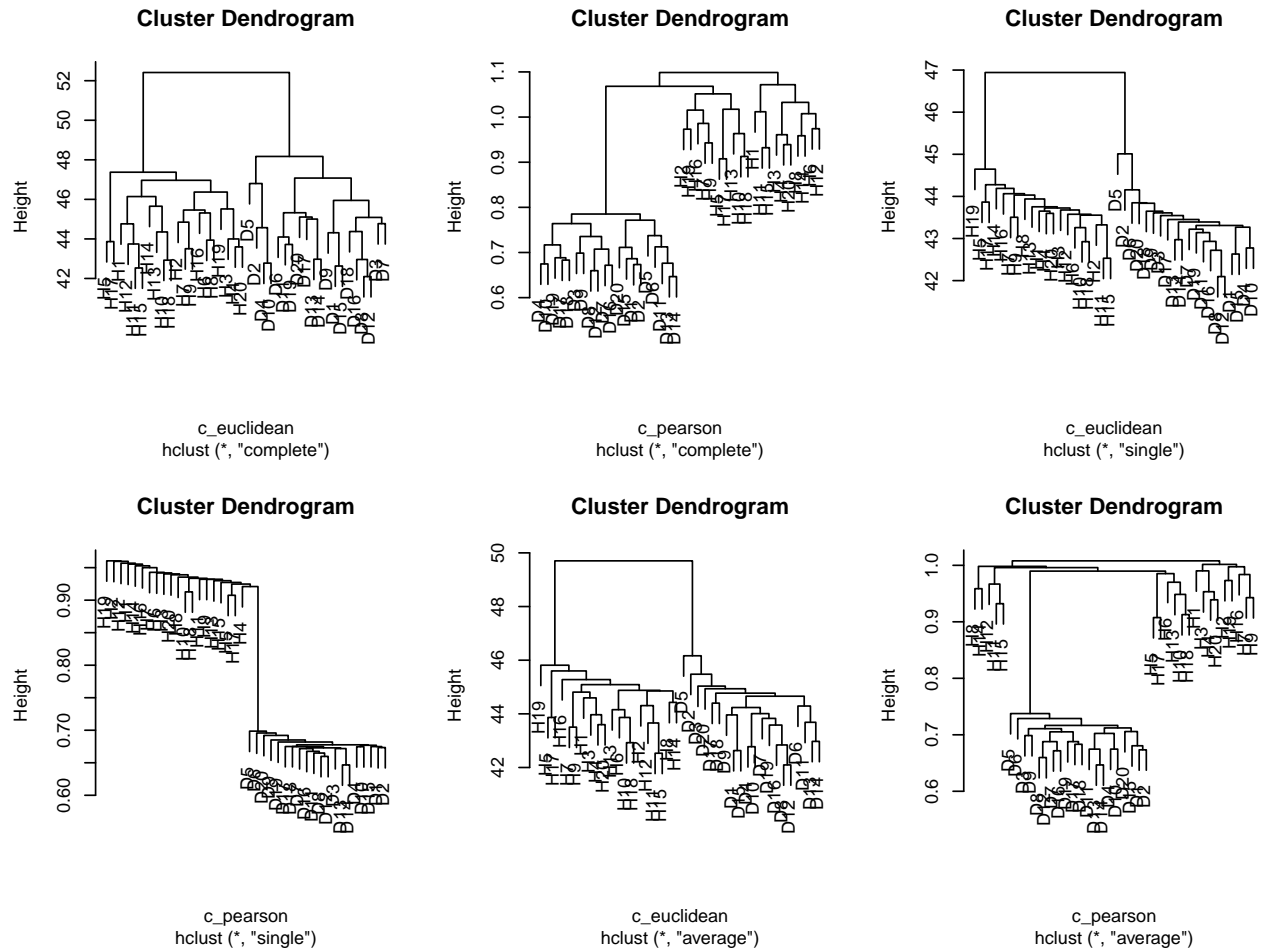
```
##  c8 nc3  c7 nc2 c18 c17  c1  c2 nc5 nc1 c13 nc6 c16  c3 c15 c10  c5 nc4  c4 c12
##   1   1   1   1   2   2   1   2   2   2   2   1   2   2   1   1   1   1   1   1
## c14  c9  c6 c11
##   1   2   1   2
```

Cluster Dendrogram



## Healthy and non-healthy tissues samples

### On R

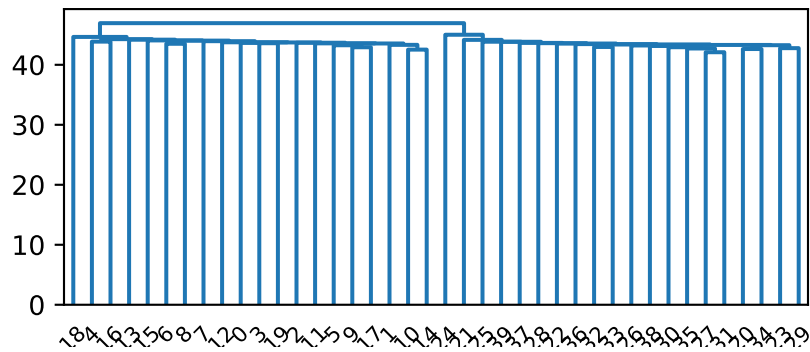


We can see that the use of euclidean distance in the three methods (complete, single, average) gives good results (no missclassification) but the use of correlation-distance gives very bad results.

Furthermore all methods, except Average with correlation-distance, divide the graph in two groups (healthy and non-healthy) which is very good.

**On python with scikit-learn** Doing hierarchical clustering with python gives the following dendrogram which shows that individual between 0 and 19 and individuals between 20 and 39 are well separated.

```
## {'icoord': [[15.0, 15.0, 25.0, 25.0], [55.0, 55.0, 65.0, 65.0], [105.0, 105.0, 115.0, 115.0], [95.0,
```



Corona dataset

## Fuzzy Clustering

Overall

TO-DO

### Examples

The function `fanny()` computes fuzzy clustering.

We use the tissue dataset.

For the dissimilarity function, the choice was euclidean distance.

2 is for the number of clusters expected.

`stand` indicates whether variables are standardized before calculating the dissimilarities.

```
## H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19 H20
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

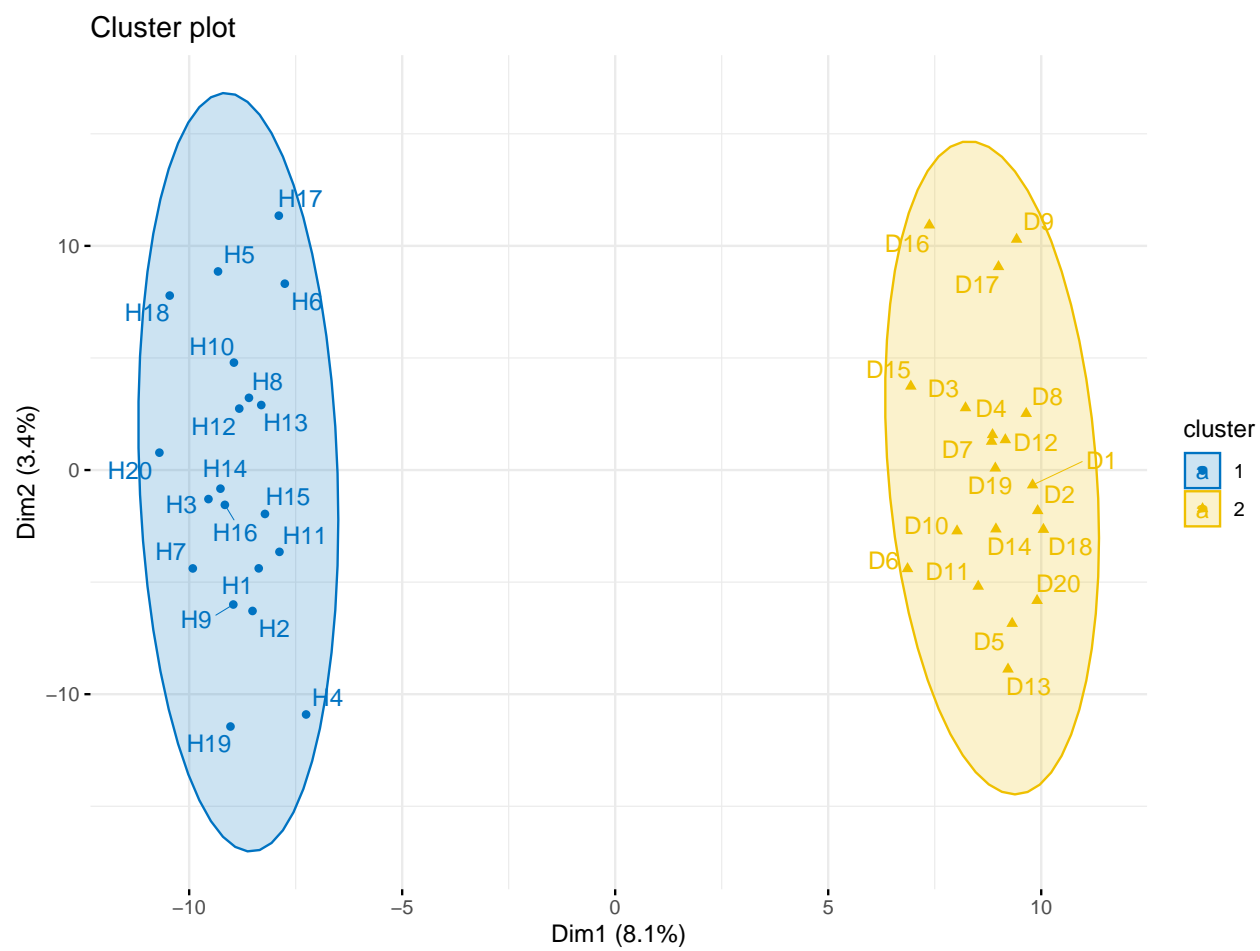
We can see that every tissue sample was well clustered.

```
## dunn_coeff normalized
## 5.000000e-01 2.220446e-15
```

Looking at the Dunn coefficient, it is about 0.5 meaning that the diameter of one of the cluster is 2 times larger than the minimal separation of two cluster.

This makes sense looking at the following plot.





## Choosing the best algorithm ?

TO-DO