Homework 2

Pattern Mining and Social Network Analysis

BOUYSSOU Gatien , de POURTALES Caroline, LAMBA Ankit

23 octobre, 2020

Contents

Continuous

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

Silhouette coefficient

The Silhouette coefficient evaluates the performance of your clustering model on a dataset. 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 :

$$Silhouette\,Score = (b - a)/max(a, b)$$

Where :

- a is the distance between each point within a cluster
- b is the distance between all the clusters

Dunn index

The dunn index is the min distance between two clusters (separation) over the max distance btw the objects of one clusters (diameter).

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

Discrete

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

Principal Components Analysis

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

Different kinds of PCA

Standard PCA

Incremental PCA

Sparse PCA

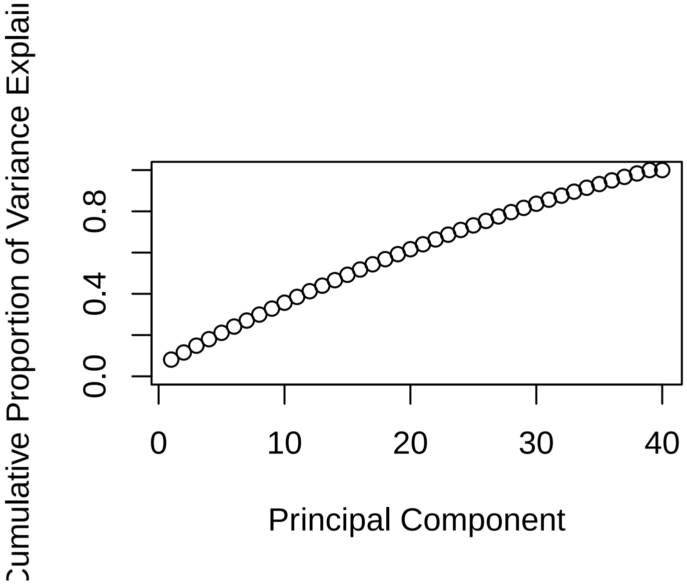
Kernel PCA

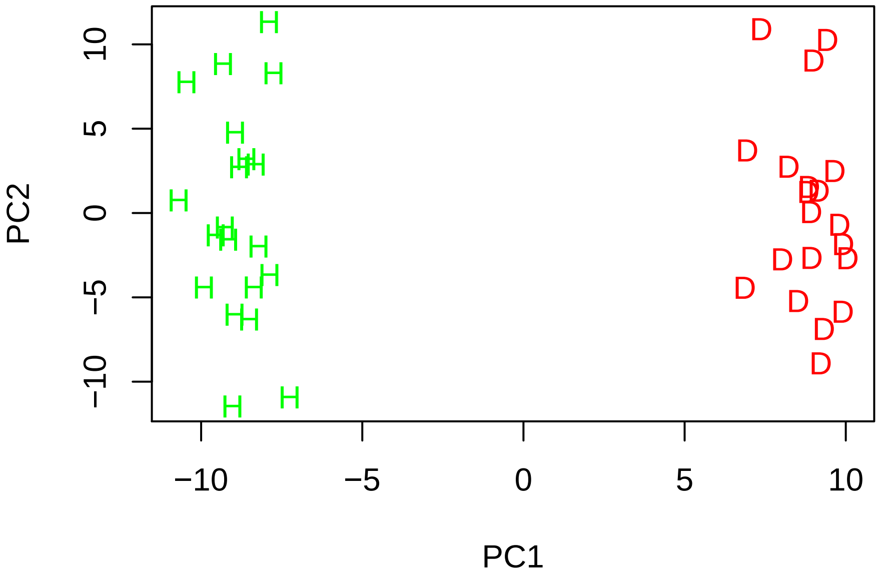
Proportion of variance explained (PVE)

Deciding how many PCs to use

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).




PC2
10
5
0
−5
−10
PC1
−10
−5
0
5
10
H
D

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 μ_k is the center of the cluster k. The total with-cluster variation is TW :

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

K-means algorithm

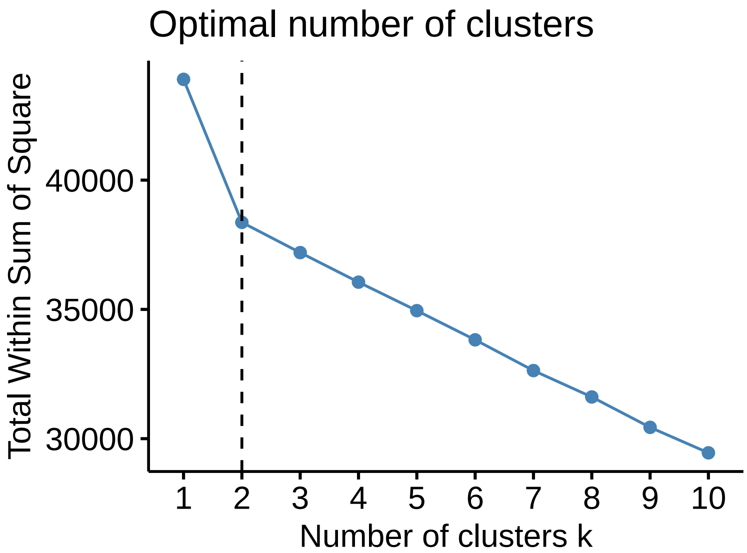
The first step when using k-means clustering is to indicate the number of clusters (k) that will be generated in the final solution. The algorithm starts by randomly selecting k objects from the data set to serve as the initial centers for the clusters. The selected objects are also known as cluster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

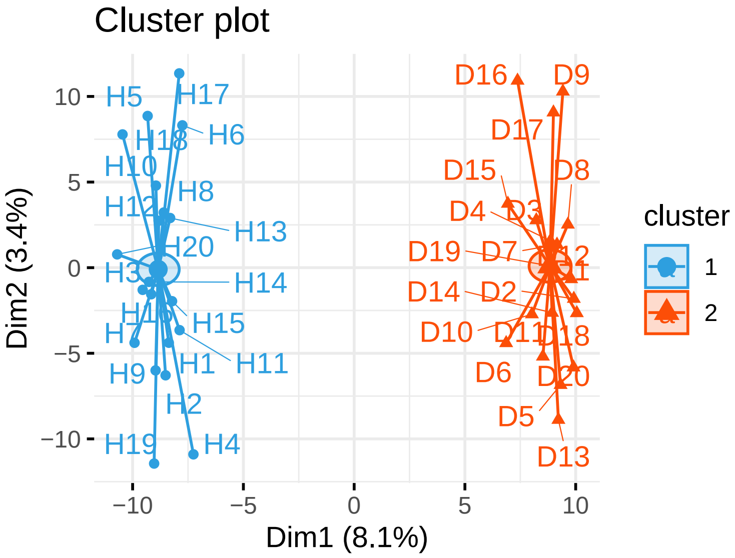
On R According to this graph, we should choose k=2 (it makes sense since we have Healthy and non healthy patients).



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 assgnation 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, 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

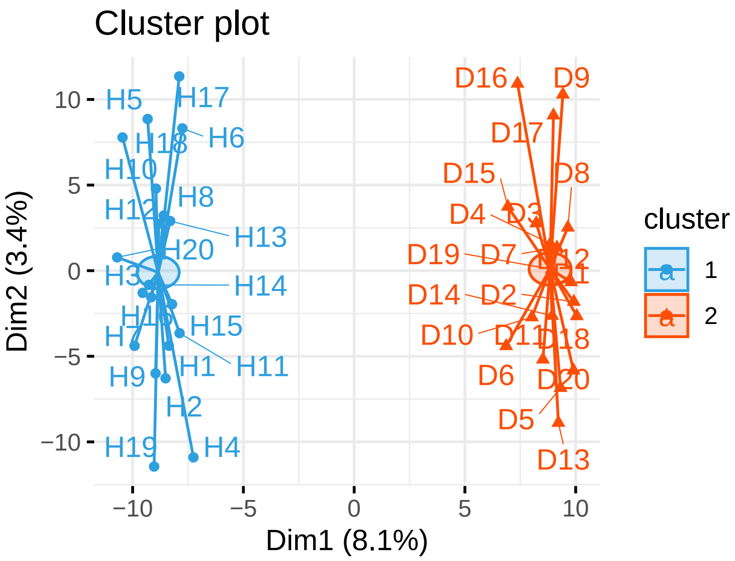
Principle

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
```


Cluster plot
Dim2 (3.4%)
Dim1 (8.1%)
10
5
0
−5
−10
−10
−5
0
5
10
H5
H17
H6
H18
H10
H8
H12
H13
H20
H3
H14
H16
H15
H7
H1
H11
H9
H2
H19
H4
D16
D9
D17
D15
D8
D4
D3
D19
D7
D12
D1
D14
D2
D10
D11
D18
D6
D20
D5
D13
cluster
1
2

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 vari- ance. At each step the pair of clusters with minimum between-cluster distance are merged.

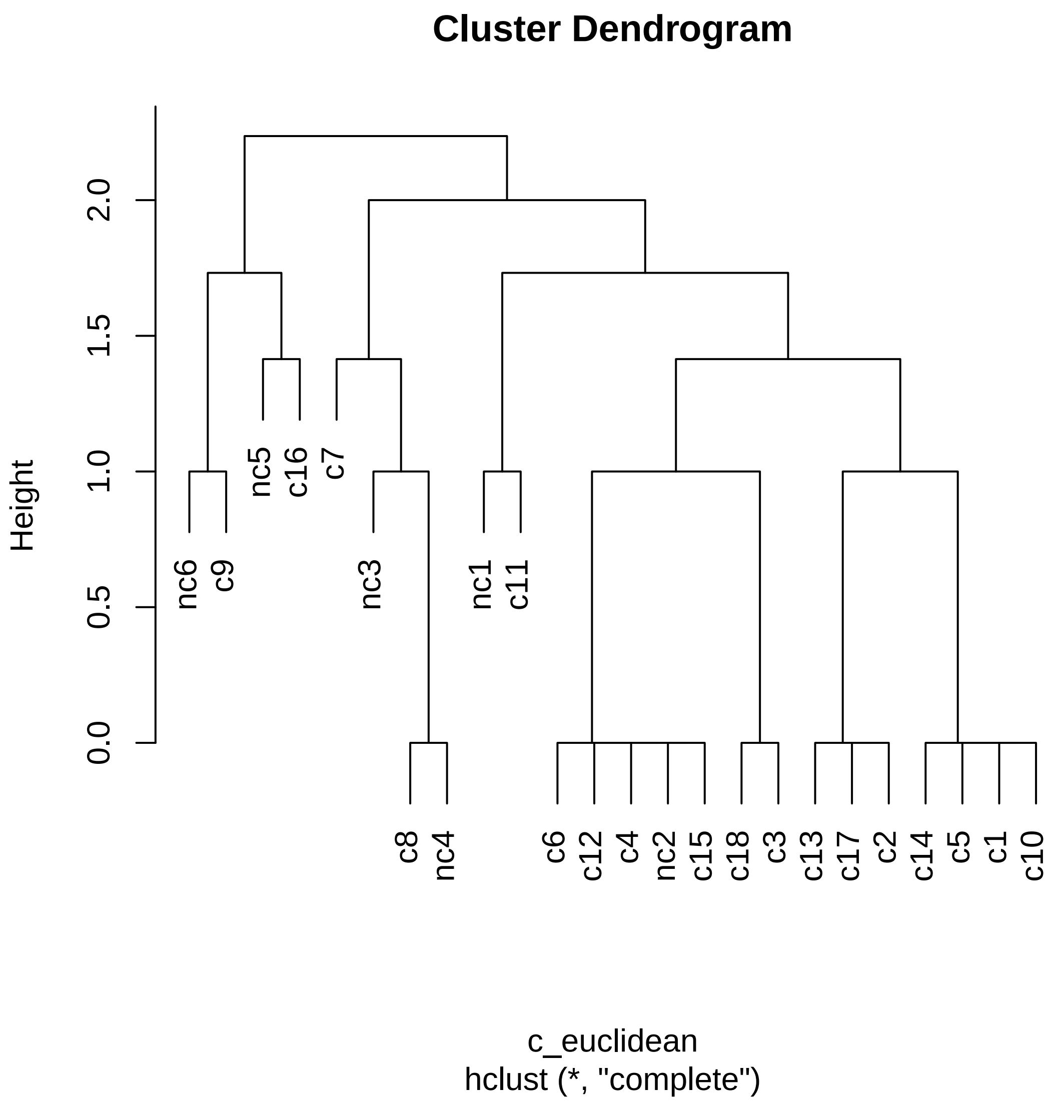
Example on mushrooms dataset

On R

```
##     Odorant Anneaux Chapeau.bombé Pied.large Tâches
## c1        1       0             1          1      0
## c2        1       0             1          1      1
## c3        0       0             1          1      1
## c4        0       0             1          1      0
## c5        1       0             1          1      0
## c6        0       0             1          1      0
## c7        1       1             0          1      0
## c8        1       1             1          1      0
## c9        0       0             1          0      1
## c10       1       0             1          1      0
## c11       0       0             0          1      1
## c12       0       0             1          1      0
## c13       1       0             1          1      1
## c14       1       0             1          1      0
## c15       0       0             1          1      0
## c16       1       1             1          0      1
## c17       1       0             1          1      1
## c18       0       0             1          1      1
## nc1       1       0             0          1      1
## nc2       0       0             1          1      0
## nc3       0       1             1          1      0
## nc4       1       1             1          1      0
## nc5       0       1             0          0      1
## nc6       0       0             1          0      0
```

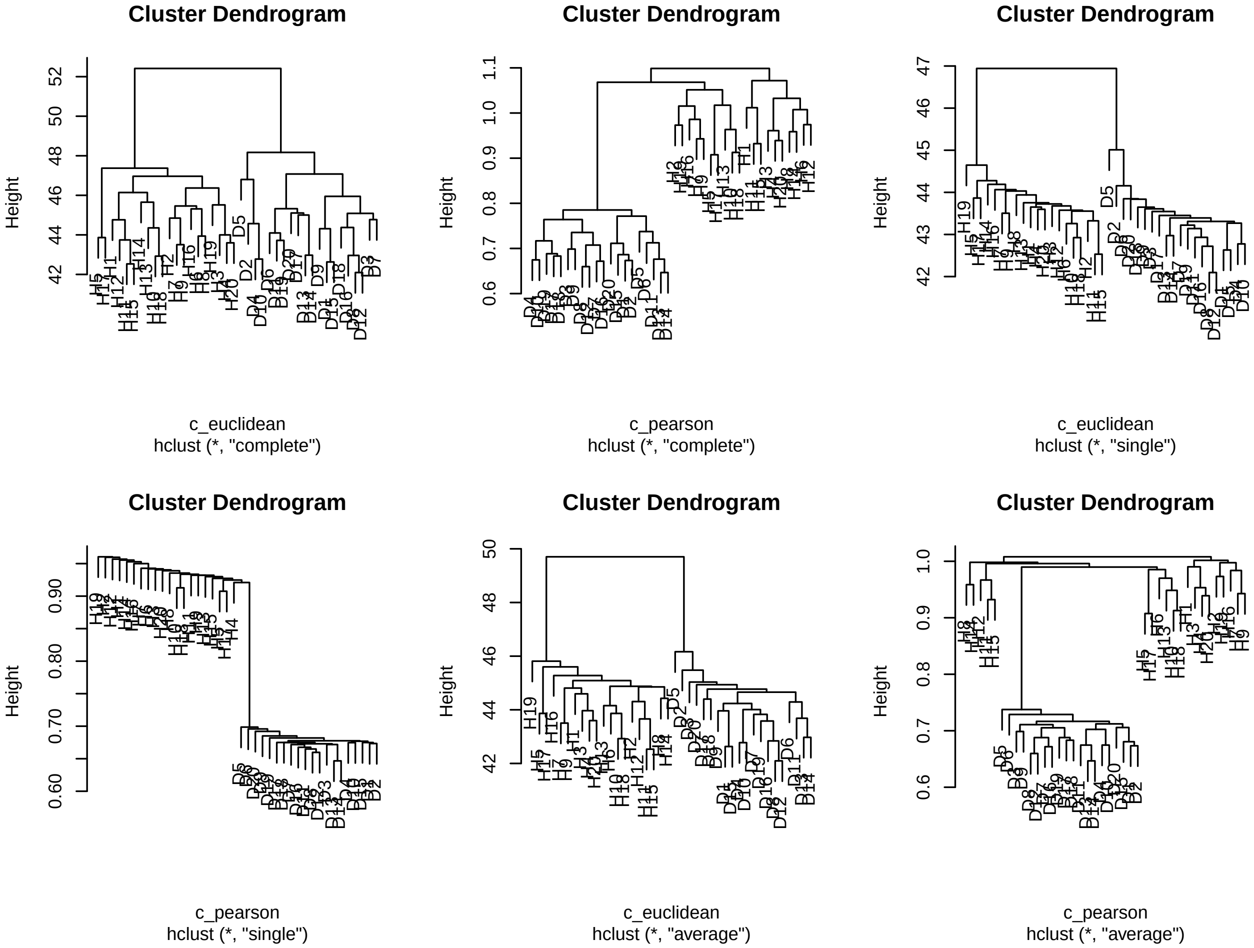
```
##  c8 nc3  c7 nc2 c18 c17  c1  c2 nc5 nc1 c13 nc6 c16  c3 c15 c10  c5 nc4  c4 c12
##   1   1   1   1   2   2   1   2   2   2   2   1   2   2   1   1   1   1   1   1
## c14  c9  c6 c11
##   1   2   1   2
```

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



Example on 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 hiearchical clustering with python gives the following dendogramm which shows that individual between 0 and 19 and individuals between 20 and 39 are well separated.

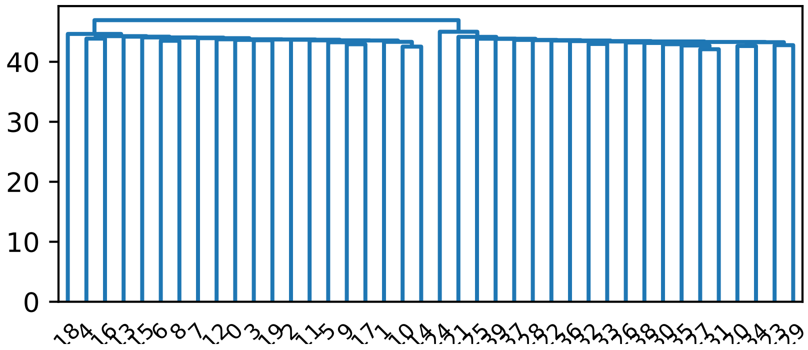
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

linked = linkage(r.genematrix, 'single')

plt.figure(figsize=(5, 2))
dendrogram(linked)
```

```
## {'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,
```

```
plt.show()
```



Example on corona dataset

```
##   H1   H2   H3   H4   H5   H6   H7   H8   H9  H10  H11  H12  H13  H14  H15  H16
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H17  H18  H19  H20  H21  H22  H23  H24  H25  H26  H27  H28  H29  H30  H31  H32
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H33  H34  H35  H36  H37  H38  H39  H40  H41  H42  H43  H44  H45  H46  H47  H48
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H49  H50  H51  H52  H53  H54  H55  H56  H57  H58  H59  H60  H61  H62  H63  H64
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H65  H66  H67  H68  H69  H70  H71  H72  H73  H74  H75  H76  H77  H78  H79  H80
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H81  H82  H83  H84  H85  H86  H87  H88  H89  H90  H91  H92  H93  H94  H95  H96
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  H97  H98  H99 H100 H101 H102 H103 H104 H105 H106 H107 H108 H109 H110 H111 H112
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H113 H114 H115 H116 H117 H118 H119 H120 H121 H122 H123 H124 H125 H126 H127 H128
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H129 H130 H131 H132 H133 H134 H135 H136 H137 H138 H139 H140 H141 H142 H143 H144
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H145 H146 H147 H148 H149 H150 H151 H152 H153 H154 H155 H156 H157 H158 H159 H160
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H161 H162 H163 H164 H165 H166 H167 H168 H169 H170 H171 H172 H173 H174 H175 H176
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H177 H178 H179 H180 H181 H182 H183 H184 H185 H186 H187 H188 H189 H190 H191 H192
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H193 H194 H195 H196 H197 H198 H199 H200 H201 H202 H203 H204 H205 H206 H207 H208
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H209 H210 H211 H212 H213 H214 H215 H216 H217 H218 H219 H220 H221 H222 H223 H224
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H225 H226 H227 H228 H229 H230 H231 H232 H233 H234 H235 H236 H237 H238 H239 H240
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H241 H242 H243 H244 H245 H246 H247 H248 H249 H250 H251 H252 H253 H254 H255 H256
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H257 H258 H259 H260 H261 H262 H263 H264 H265 H266 H267 H268 H269 H270 H271 H272
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H273 H274 H275 H276 H277 H278 H279 H280 H281 H282 H283 H284 H285 H286 H287 H288
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H289 H290 H291 H292 H293 H294 H295 H296 H297 H298 H299 H300 H301 H302 H303 H304
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H305 H306 H307 H308 H309 H310 H311 H312 H313 H314 H315 H316 H317 H318 H319 H320
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H321 H322 H323 H324 H325 H326 H327 H328 H329 H330 H331 H332 H333 H334 H335 H336
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H337 H338 H339 H340 H341 H342 H343 H344 H345 H346 H347 H348 H349 H350 H351 H352
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H353 H354 H355 H356 H357 H358 H359 H360 H361 H362 H363 H364 H365 H366 H367 H368
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H369 H370 H371 H372 H373 H374 H375 H376 H377 H378 H379 H380 H381 H382 H383 H384
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H385 H386 H387 H388 H389 H390 H391 H392 H393 H394 H395 H396 H397 H398 H399 H400
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H401 H402 H403 H404 H405 H406 H407 H408 H409 H410 H411 H412 H413 H414 H415 H416
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H417 H418 H419 H420 H421 H422 H423 H424 H425 H426 H427 H428 H429 H430 H431 H432
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H433 H434 H435 H436 H437 H438 H439 H440 H441 H442 H443 H444 H445 H446 H447 H448
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## H449 H450   D1   D2   D3   D4   D5   D6   D7   D8   D9  D10  D11  D12  D13  D14
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  D15  D16  D17  D18  D19  D20  D21  D22  D23  D24  D25  D26  D27  D28  D29  D30
##    1    1    1    1    1    2    2    2    2    2    2    2    2    2    2    2
##  D31  D32  D33  D34  D35  D36  D37  D38  D39  D40  D41  D42  D43  D44  D45
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
```

Validation techniques

Bootstrapping

Example

On R

On python with scikit-learn