MALDITOFSpectraPA

Sautié Castellanos, Miguel

miguel.sautie.castellanos@umontreal.ca
msc91007@gmail.com
https://github.com/Sautie/MALDITOFSpectraPA

MALDI_TOFSpectraPA is a set of 22 R functions for MALDI_TOF spectra processing and analysis. These functions are based on built-in functions and other functions coming from different packages for statistical multivariate analysis as well as MALDI_TOF spectra import and processing. These functions can be grouped into 6 classes, as shown below.

1 Metadata processing: AgFilter, PMetaData, MergeMetaData, DropCMetadata

2 MALDI_TOF Spectrum processing: CopyRenameXml, XmlNCleaning, SlowXml26Cleaning, Peak_Detector, RowColumnSelector

3 Clustering of MALDI_TOF Spectra and cluster validation: BHclus, Phyclus, PhyclusVar, Vizclus, Dendrogram_pairComp, OptClusters, VisualOptClusters, PointClusterVal

4 Correlograms for MALDI_TOF spectra: Visual_CorrDistM

5 Principal Component Analysis and clustering of MALDI_TOF spectra: PCA_Clus, SPCA

6 Multidimensional scaling and clustering of MALDI_TOF spectra. MDS_Clus, SMDS

These functions are organized in the following 8 R scripts.

source("MetaDProcessing.R"), source("DProcessing.R"), source("SProcessing.R"),
source("HClus_dendrograms.R"), source("ClusVal.R"), source("VisualDM_Correlograms.R")
source("S_PCA_Clus.R"), source("DClus_MS.R")

and depend on the following 18 R packages referred below,

library("MALDIquant"), library("MALDIquantForeign"), library("matrixStats"), library("stringr")
library("readxl"), library("(tidyverse"), library("dplyr"), library("readxl"), library("factoextra")
library("fpc"), library("cluster"), library("ggplot2"), library("FactoMineR"), library("corrplot")
library("ggpubr"), library("pvclust"), library("ape"). library("dendextend")

The functions were included in the R package MaldiTOFSpectraPA_0.1.0 (Source and binary packages delivered together with this document).

The development of this small set of functions has two basic objectives: 1) Accelerating as much as possible the processing of spectra and metadata contained in csv, xlsx and mzXML files and directories. 2) developing pipelines that connect these processing functions to a few of the most known R functions for clustering, PCA, Classic Multidimensional Scaling, correlograms and dendrogram building. Most of the arguments of these functions come directly from the functions they use internally, and, in many cases, are the most relevant ones used by these functions. Thus, probably by adjusting only the values of these arguments, the desired graphs and data displays can be obtained. The rest of the arguments of these internal functions take either default values, or values set according to the specific characteristics of these data or the concrete examples taken from the literature.

In this document we describe the general characteristics of each function. This description of each function is structured in four sections. In section @param, the arguments of each function are presented as well as the values they take and the default values. In the sections identified as @return and @examples, as their names indicate, the returned values and concrete examples of use of each function are shown. The last section, called source, contains links to documents that were very useful for the development of the corresponding function. Below the description of each function, some examples of its use are shown in much more detail.

The output folders and files of the most relevant functions for the first two groups of functions linked to spectra and metadata processing, are compressed together with the R scripts in zip files.

Table des matières

1. Metadata processing

1.1. AgFilter

AgFilter is an internal function for column selection ("select" , default value) or column aggregation based on "means", "medians", "max" and "min" of scores.

1.2. PMetaData

PMetaData extracts from csv files the taxonomic identifications of the isolates and transforms them according to the values of the scores

@param Dirp: folder (and path) where the csv files are located,
@param fileout: name of the output file containing the Maldi identifiers and species names,
@param fu: this parameter determines whether only one of the scores (y, default value=2) is considered, or inste ad, the mean, median, minimum or maximum of the scores corresponding to each of the isolates, "means", "medians", "select", "max" and "min",
@param id: determine whether unidentified isolates are labeled with "NRI" (default value) or removed,
@param sc: columns containing the scores
@param spNames: columns containing the names of the identified species
@return dataframe (and csv file) with three columns: Maldi identifiers, isolate identification s and the corresponding scores
@examples Rn<-PMetaData("Bees project 2019-2020"),
Rn<-PMetaData("Bees project 2019-2020", fileout="expMeans.csv", fu="means")
Rn<-PMetaData("Bees project 2019-2020", fileout="expMedians.csv", fu="medians")
Rn<-PMetaData("Bees project 2019-2020", fileout="expMin.csv", fu="min"),
Rn<-PMetaData("Bees project 2019-2020", fileout="expMax.csv", fu="max")

1.3. MergeMetaData

It merges two dataframes, one is the output of the function PMetaData1 and the other comes from a xlsx file. The merging is done through the Maldi code.

@param df:a dataframe which is an output of PMetaData1,
@param Dp: folder containing the metadata files,
@param filein: xlsx files for metadata
@param fileout: csv file for both joined dataframes,
@param keyCode: key code used for Dataframe merging, "Maldi_code" (Default value)
@return both dataframes joined into one dataframe and exported as an csv file
@examples dfm<-MergeMetaData(Rn, "Bees project 2019-2020","Bees metadata.xlsx", "Filered_Meta.csv")

1.4. DropCMetadata

DropCMetadata removes previously chosen variables from the metadata csv files

@param dm: dataframe,
@param fileout: csv file for both joined dataframes
@return output dataframe and csv file
@examples dfc<-DropCMetadata(dfm,"DCFilered_Meta.csv", c("Date d'analyse","Date de récolte"))

2. MALDI_TOF Spectrum processing

2.1. CopyRenameXml

CopyRenameXml is a function aimed at copying and renaming the mzXML files not being in the folders: "BTS", "BTS_Validation", "CTL", "Autocalibration"

@param Dirp: folder containing the mzXML files,
@param newdir: the folder to which the mzXML files are copied
@examples CopyRenameXml("Bees project 2019-2020", "newdata")

2.2. XmlNCleaning

XmlNCleaning copies all mzXML files with at least n lines to a new folder or delete those with less than n lines, (the mzXML files have 26 lines), this way incomplete or truncated files are removed...the number of characters of line n=18 is also verified in order to detect other defective files.

@param Dirp: folder containing the Maldi_Tof spectrum files,
@param newdir: folder where the chosen files are copied,
@param op: "delete"(default value) or "copy",
@param n: number of lines (n=18 default),
@param number: of characters of line 8 (number=440518, default value)
@examples XmlNCleaning("newdata"),
XmlNCleaning("newdata5", "newdata", op="copy"),
XmlNCleaning("newdata5", "newdata", op="copy", n=26)

Note: it's recommended to run this function with op="delete" to clean the CopyRenameXml output directory

2.3. SlowXml26Cleaning

SlowXml26Cleaning copies all mzXML files with 26 lines to a new folder or delete those with less than 26 lines, slow **and** old version of XmlNCleaning

@param Dirp: folder containing the Maldi_Tof spectrum files,
@param newdir: folder where the chosen files are copied,
@param op: "delete"(default value) or "copy"
@examples SlowXml26Cleaning ("newdata5", "newdata"),
SlowXml26Cleaning ("newdata5", "newdata", op="copy")

2.4. Peak_Detector

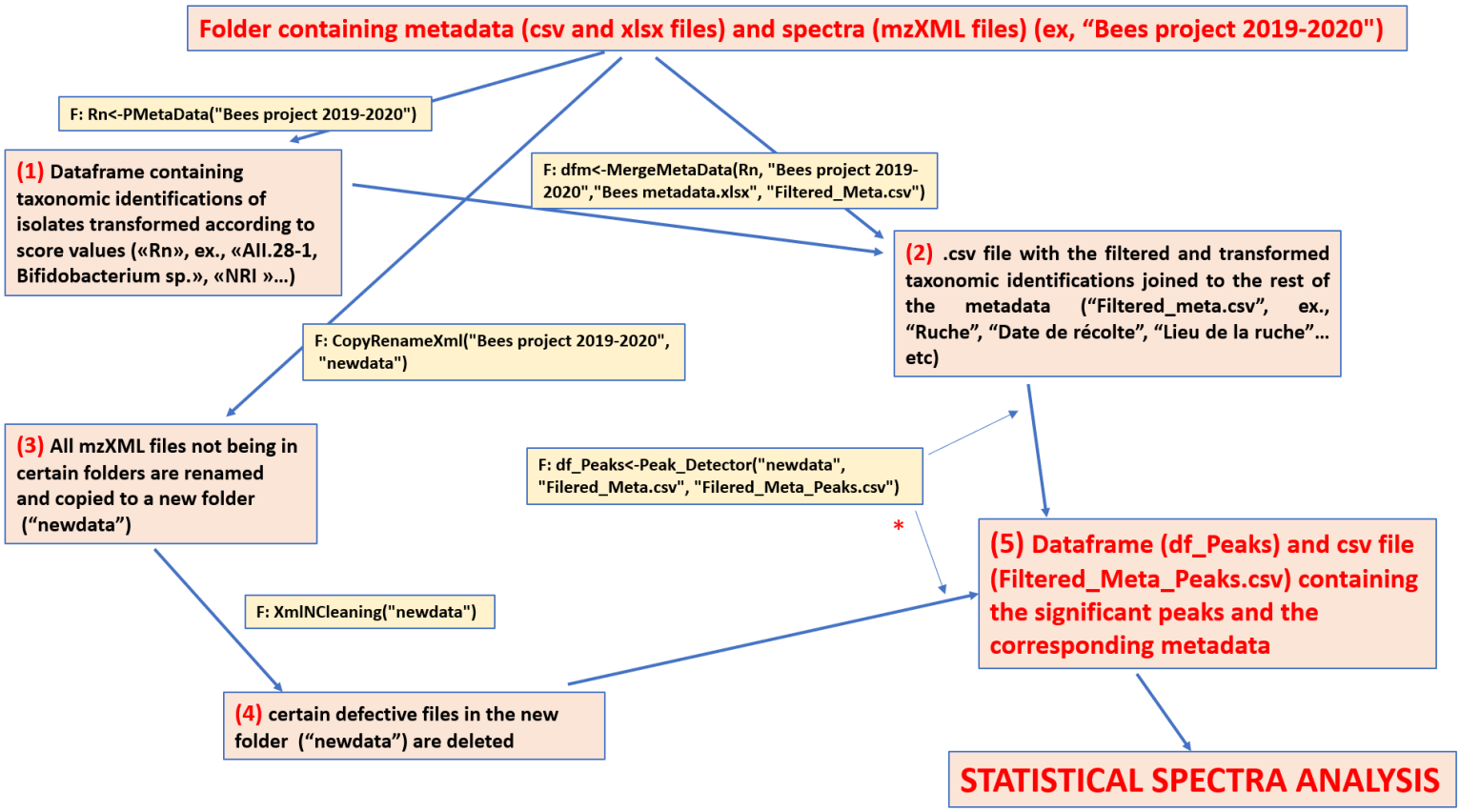
Peak_Detector is a pipeline of two stages for 1) importing and processing of Maldi_Tof spectra and 2) peak detection:

Stage 1) import, check of quality, transformation, smoothing, baseline removing, normalization, and alignment of Maldi_Tof spectra.
Stage 2) peak detection, binning, filtering and merging with metadata.

@param Dp: folder containing the ".mzXML" files,
@param filein: metadata input csv file,
@param fileout: output csv file,
@param keyCode: code for dataframe joining (default:"Identifiant_MALDI"),
@param speColumn: taxonomic identification of isolates

@param t: spectrum transformation (default value, t="sqrt"), "log",
@param smooth: smoothing method (default value smooth="SavitzkyGolay"), "MovingAverage", "WMovingAverage",
@param baseline: baseline removing method, (default value, baseline="SNIP"), "TopHat", "ConvexHull",
@param normalization: calibration or normalization algorithms (default value, normalization="TIC"), "PQN",
@param Iter: number of iterations for baseline removing (default value, Iter=100)
@param SN_R: signal_to_noise ratio (default value, SN_R=2),
@param minFreq: the minimum peak frequency for spectrum selection (default value, minFreq=0.25),
@param align: Boolean parameter to indicate whether or not spectrum alignment is performed (default value, align=TRUE)
@return merged dataframes and csv file
@examples df_Peaks<-Detect_Peaks("newdata", "Filered_Meta.csv", "Filered_Meta_Peaks.csv"),
df_Peaks_0<-Detect_Peaks("newdata", "Filered_Meta.csv", "Filered_Meta_Peaks_0.csv", minFreq=0, align=FALSE)
df_PeaksTH<-Detect_Peaks("newdata", "Filered_Meta.csv", "Filered_Meta_Peaks.csv", baseline="TopHat")

note: Peak_Detector produces the dataframe (and csv file) that serves as starting point for the analysis stage. df_Peaks, (exported as Filtered_Peaks_Meta.csv), see figure below

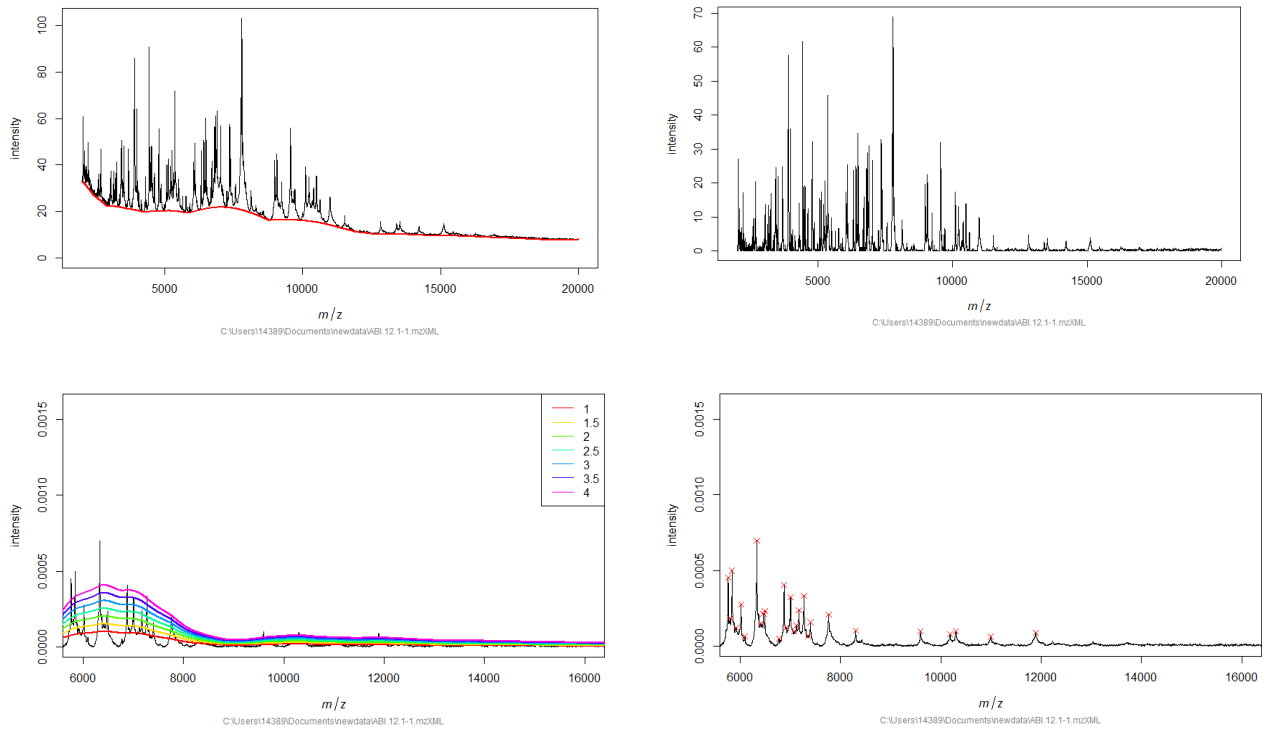


This figure shows the workflow for importing and processing MALDI_TOF spectra and metadata. The folders, csv files and dataframes generated in each of the stages of this workflow are compressed together with the scripts in the zip files. The df_Peaks dataframe groups all significant spectrum peaks and variables used in the analysis stage. The salmon-colored boxes represent the data (and metadata) and the yellow ones, the functions. *note: Peak_Detector must be executed after folder cleaning with XmlNCleaning (in the example above, the name of this folder is "newdata")

2.5. RowColumnSelector

RowColumnSelector selects rows and columns that meet certain conditions determined by a categorical variable.

@param df_m: dataframe containing peaks and metadata
@param varCat1: selected categorical variable
@param value: chosen level of varCat1
@param ni,nf: first and last columns corresponding to categorical variables
@examples filtered<-RowColumnSelector(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus")



This figure shows some of the stages performed by the Peak_Detector pipeline, baseline estimating and deleting, signal-to-noise ratio and significant peak estimates

3. Clustering of MALDI_TOF spectra and cluster validation

3.1. BHclus

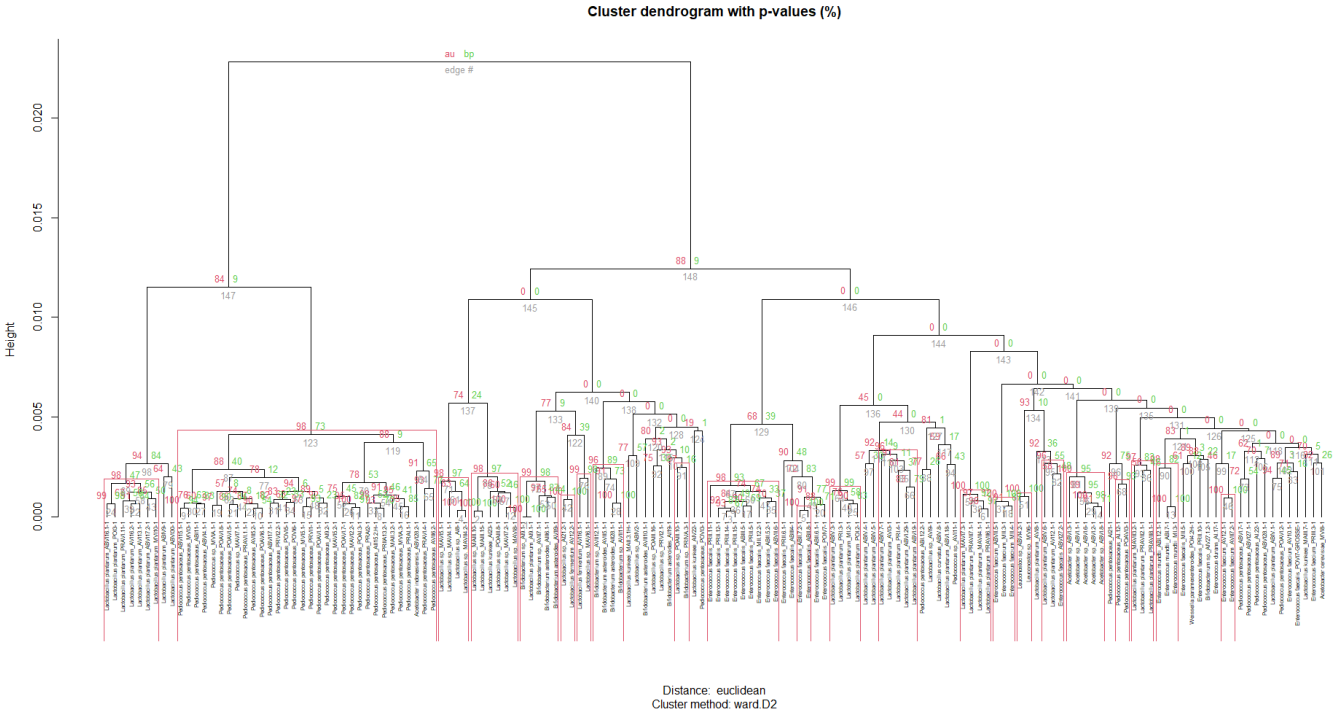
It's a function wrapper of pvclust for building dendrograms labeled with bootstrap probabilities for isolates chosen according to a categorical variable

@param df_m: dataframe containing peaks and metadata
@param meth: hierarchical clustering algorithm ("ward2", default), other values: "average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param dist: distance ("euclidean", defaul), "maximum", "manhattan", "canberra", "binary" "correlation", "uncentered", "abscor"
@param varCat1: categorical variable, "Genre","Taxonomie", " Nutrition", "Ruche"...
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param nb: number of bootstrap iterations (nb=100, default value)
@param fig: boolean variable to indicate output figure
@return output cluster and figure
@examples dft<-BHclus(df_Peaks, varCat1="Genre", value="All", nb=500), dft<-BHclus(df_Peaks, varCat1="Genre", value="Lactobacillus", nb=500)
dft<-BHclus(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus"),
dft<-BHclus(df_Peaks, meth="complete", dist="canberra",varCat1="Taxonomie", value="Pediococcus pentosaceus")

source: https://academic.oup.com/bioinformatics/article/22/12/1540/207339
https://www.rdocumentation.org/packages/pvclust/versions/2.2-0/topics/pvclust
https://www.rdocumentation.org/packages/stats/versions/3.2.1/topics/dist

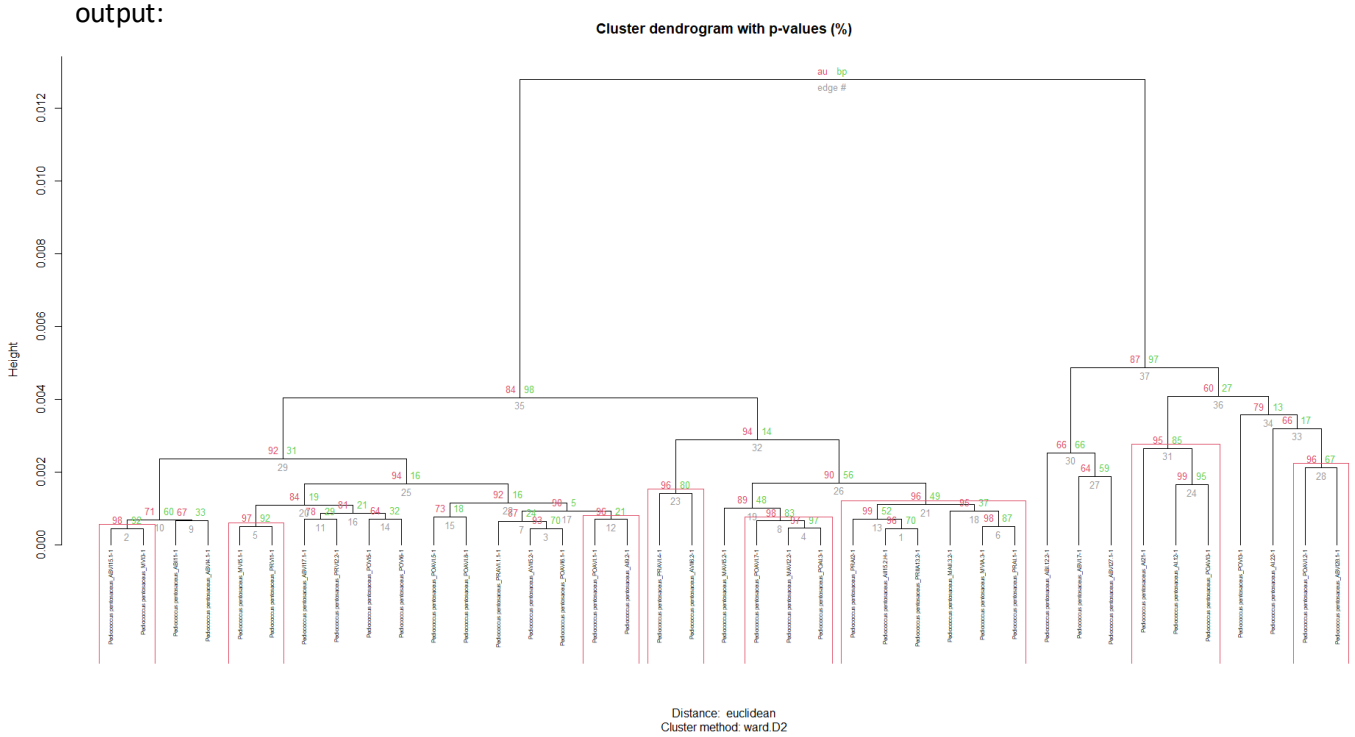
dft<-BHclus(df_Peaks, varCat1="Genre", value="All", nb=500)

output:



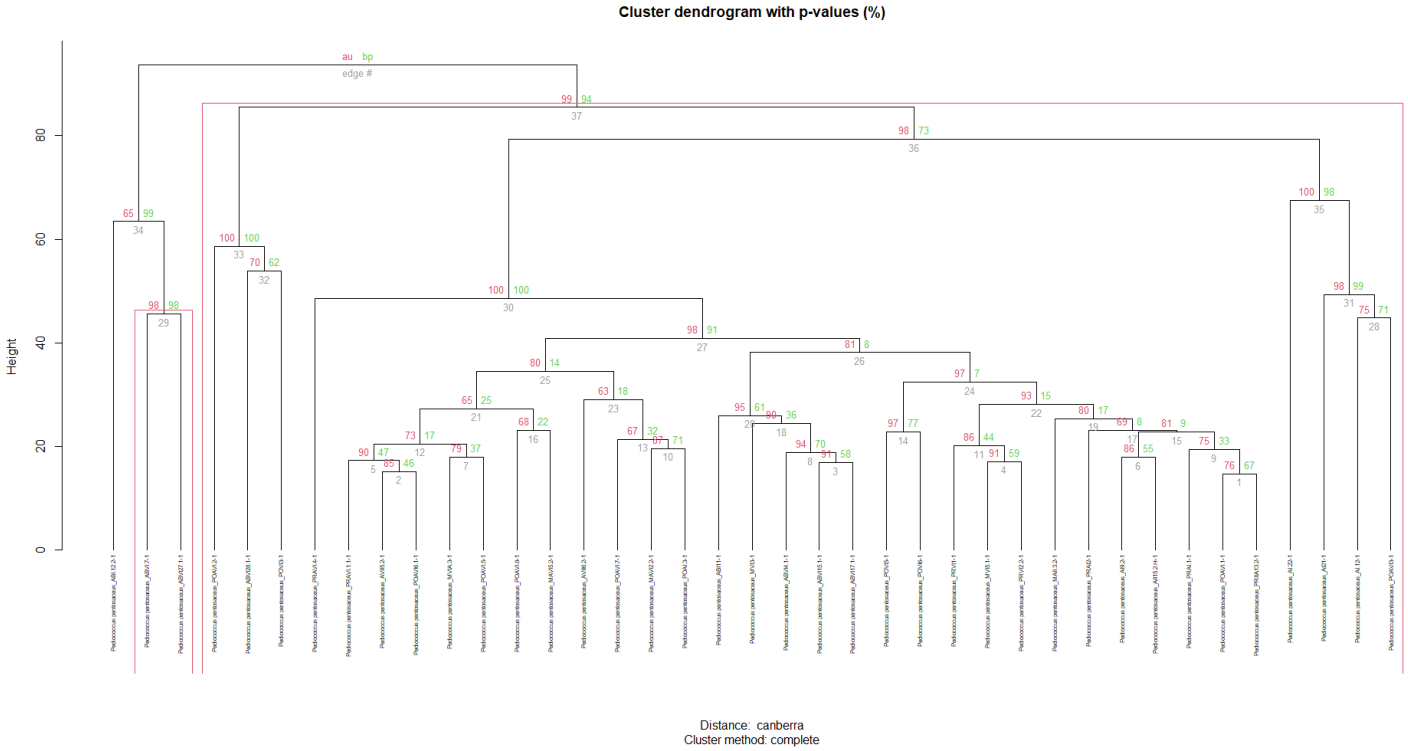
dft<-BHclus(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus")

output:



dft<-BHclus(df_Peaks, meth="complete", dist="canberra",varCat1="Taxonomie", value="Pediococcus pentosaceus")

output:



3.2. Phyclus

Phyclus builds different types of dendrograms for isolates chosen according to a categorical variable

@param df_m: dataframe containing peaks and metadata
@param meth: hierarchical clustering algorithm ("ward2", default), other values: "average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param dist: distance ("euclidean", defaul), "euclidean", "maximum", "manhattan", "canberra", "binary" "minkowski"
@param varCat1: categorical variable, example: "Genre","Taxonomie", " Nutrition", "Ruche"...
@param value: level of catVar1, example: "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param nc: number of clusters (nc=4, default value)
@param dendrogram: dendrogram type ("phylogram", default value), "cladogram", "unrooted", "fan" and "radial"
@return dataframe and figure
@examples c<-Phyclus(df_Peaks, varCat1="Taxonomie",value="Pediococcus pentosaceus"), c<-Phyclus(df_Peaks, varCat1="Nutrition", value="Erica cinerea")
c<-Phyclus(df_Peaks, varCat1="Taxonomie",value="Pediococcus pentosaceus", dendogram="cladogram")

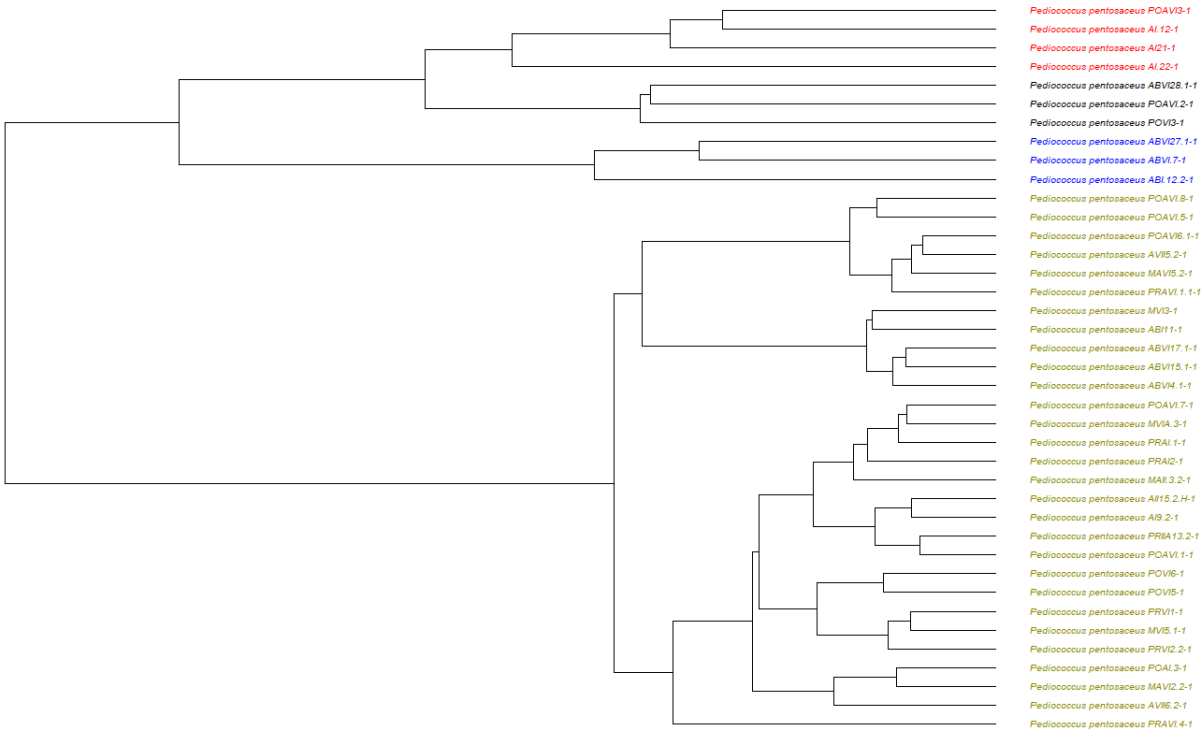
source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust
https://www.rdocumentation.org/packages/ape/versions/5.4-1

c<-Phyclus(df_Peaks, varCat1="Taxonomie",value="Pediococcus pentosaceus")

output:

[1] "Cophenetic coefficient"

[1] 0.9074493

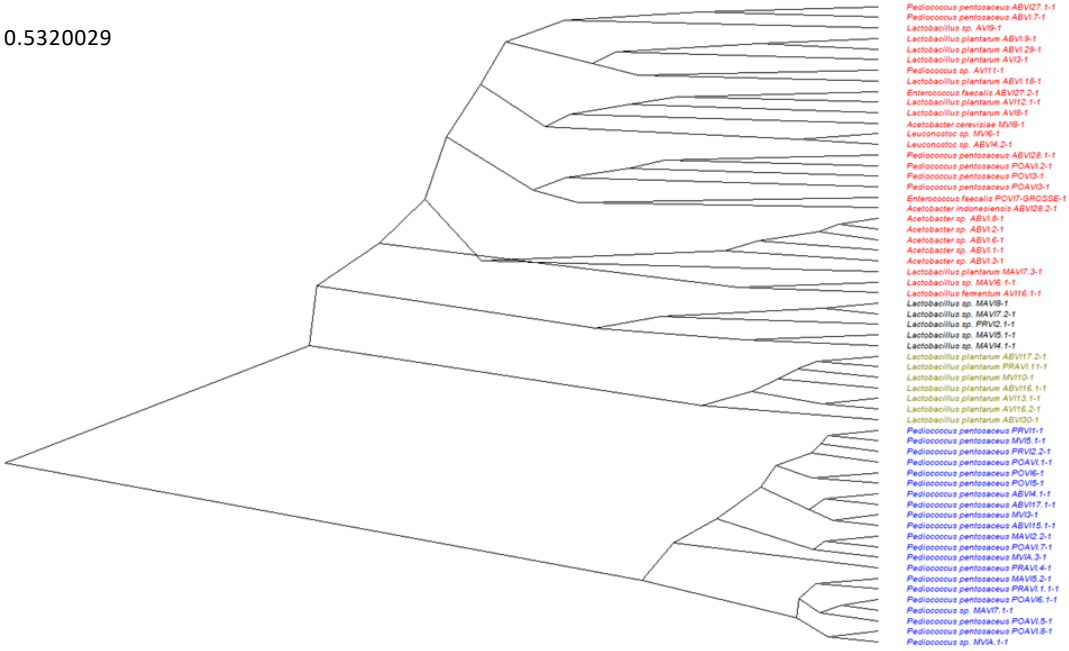


c<-Phyclus(df_Peaks, varCat1="Ruche",value="VI", dendrogram="cladogram")

output:

[1] "Cophenetic coefficient"

[1] 0.5320029

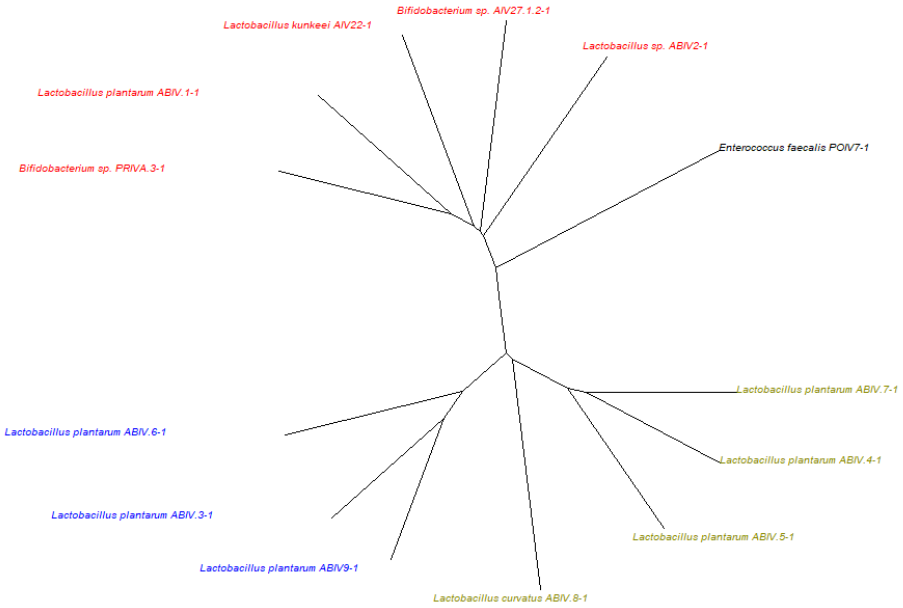


c<-Phyclus(df_Peaks, varCat1="Ruche",value="IV" , dendrogram="unrooted")

output:

[1] "Cophenetic coefficient"

[1] 0.647521

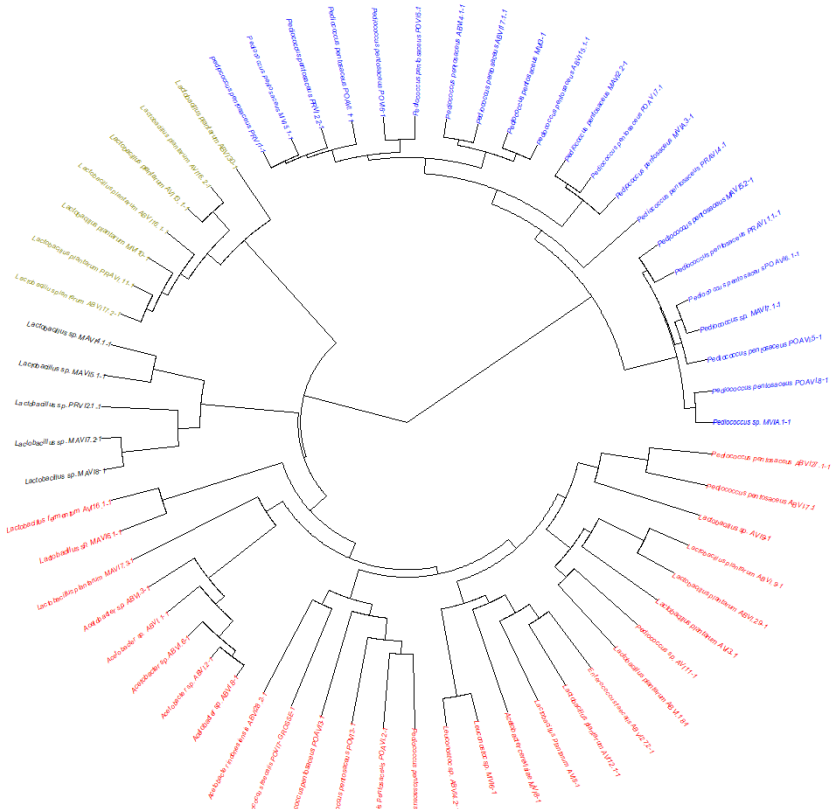


c<-Phyclus(df_Peaks, varCat1="Nutrition",value="Erica cinerea" , dendrogram="fan")

output

[1] "Cophenetic coefficient"

[1] 0.5320029

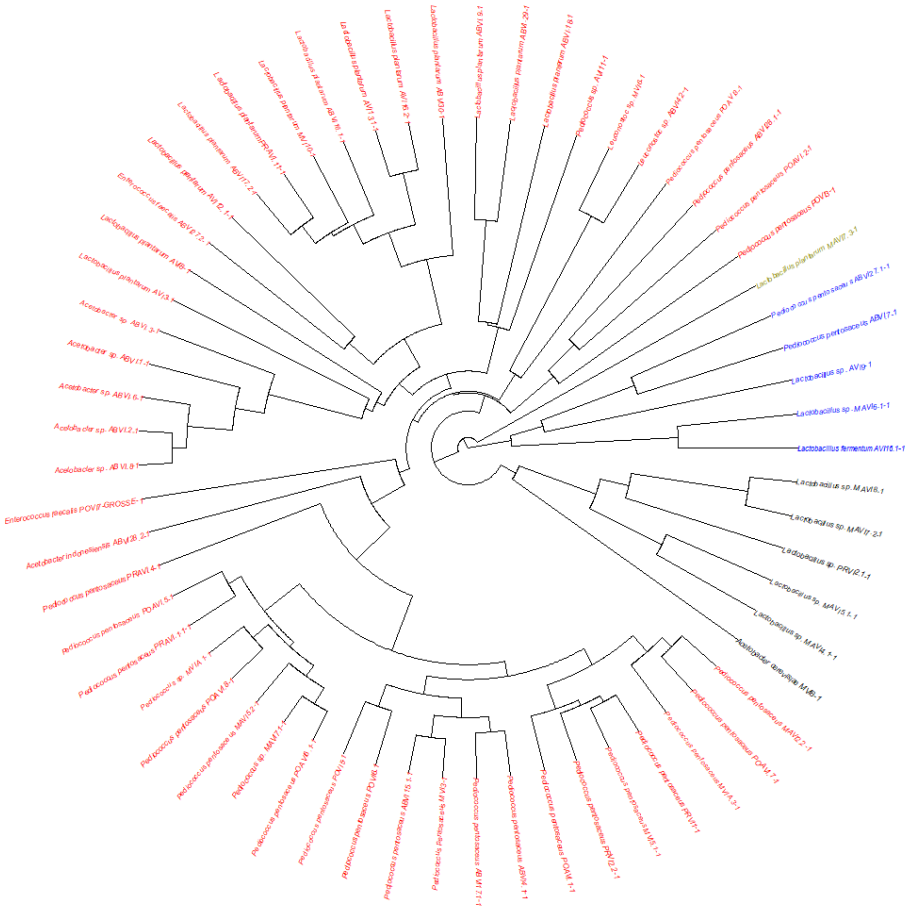


c<-Phyclus(df_Peaks, meth="complete", varCat1="Nutrition",value="Erica cinerea" , dendrogram="fan")

output:

[1] "Cophenetic coefficient"

[1] 0.9295954



3.3. PhyclusVar

PhyclusVar builds different types of dendrograms for isolates selected and categorized based on varCat2 and varCat1 levels

@param df_m: dataframe containing peaks and metadata
@param meth: hierarchical clustering algorithm ("ward2", default value), other values: "average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param dist: distance ("euclidean", default value), "euclidean", "maximum", "manhattan", "canberra", "binary" "minkowski"
@param varCat1: categorical variable, examples: "Taxonomie", "Genre", "Date.d.analyse", "Origine","Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param varCat2: categorical variable
@param nc: number of clusters (nc=4, default value)
@param dendrogram: dendrogram type ("phylogram", default value), "cladogram", "unrooted", "fan" and "radial"
@return output cophenetic coefficient and figure
@examples dft<-PhyclusVar(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition"),
dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="Lactobacillus plantarum", varCat2="Ruche")
dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", varCat2="Nutrition", dendrogram="cladogram"),
dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="All", varCat2="Date.de.récolte", dendrogram="fan")
dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="Lactobacillus plantarum", varCat2="Ruche", dendrogram="fan"),
dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="Lactobacillus plantarum", varCat2="Ruche", dendrogram="unrooted")

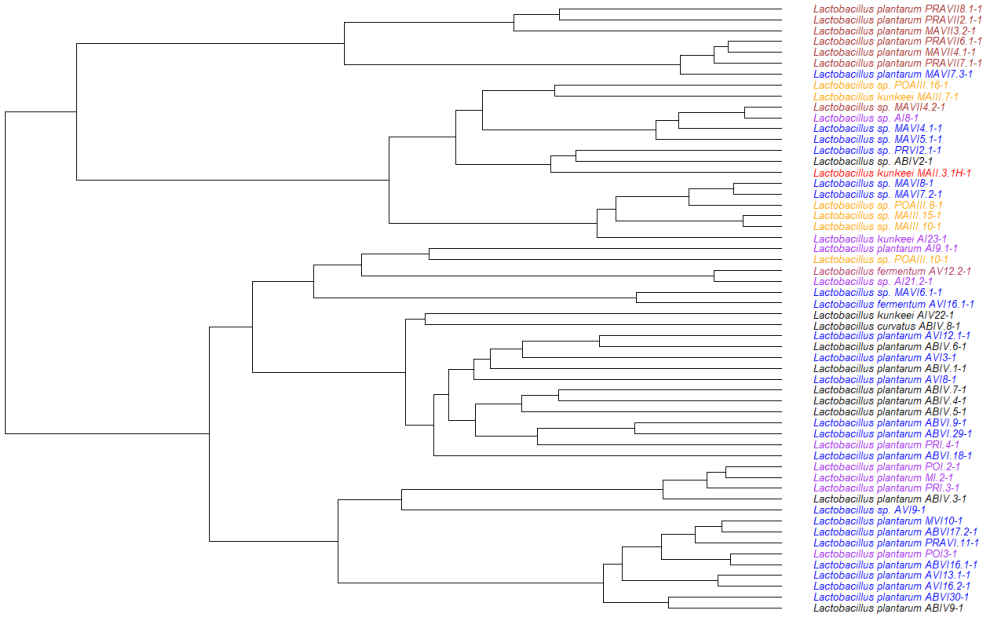
source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust
https://www.rdocumentation.org/packages/ape/versions/5.4-1

dft<-PhyclusVar(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")

output:

[1] "Cophenetic coefficient"

[1] 0.6082436

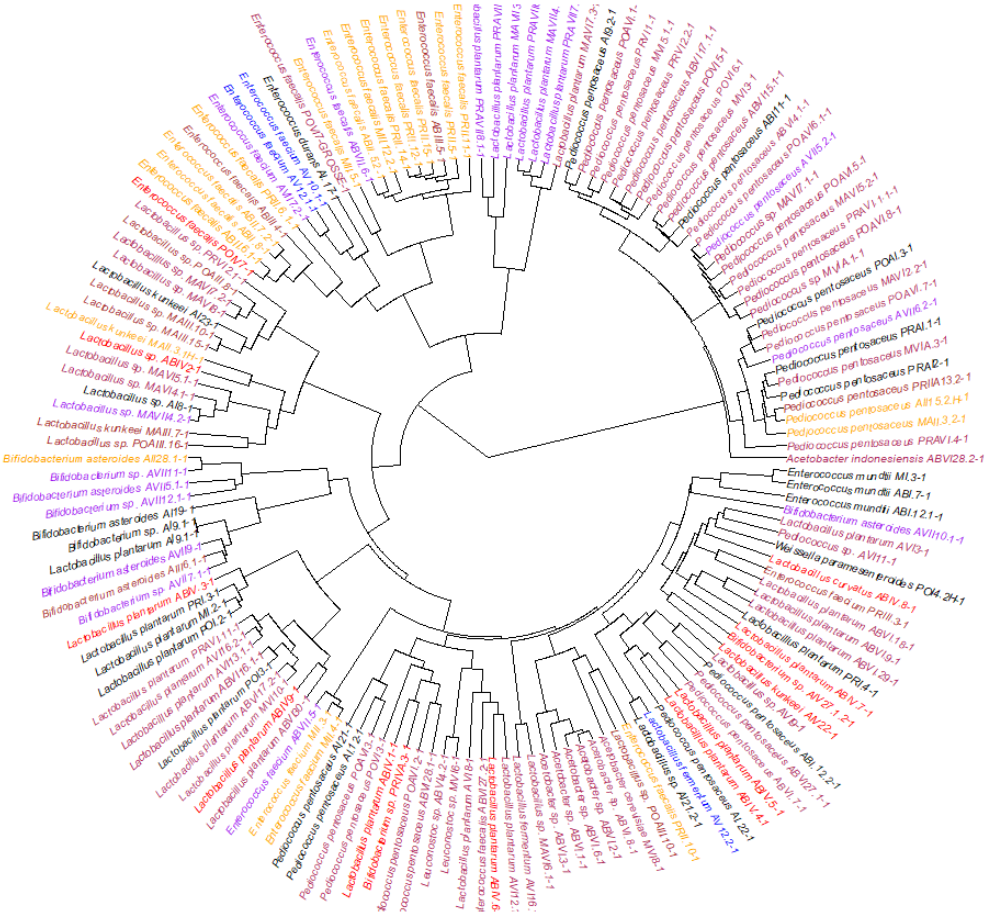


dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="All", varCat2="Date.de.récolte", dendrogram="fan")

output

[1] "Cophenetic coefficient"

[1] 0.39633

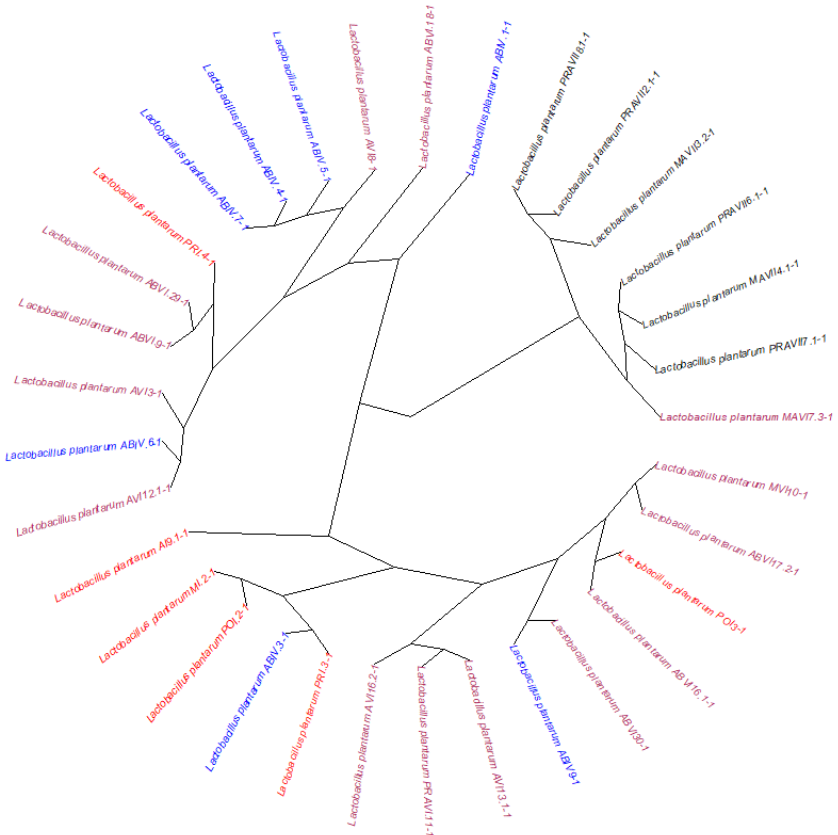


dft<-PhyclusVar(df_Peaks, varCat1="Taxonomie", value="Lactobacillus plantarum", varCat2="Ruche", dendrogram="radial")

output:

[1] "Cophenetic coefficient"

[1] 0.7178863



3.4. Vizclus

Vizclus computes clustering statistics and has other options for dendrogram visualization

@param df_m: dataframe containing peaks and metadata
@param meth: hierarchical clustering algorithm ("ward2", default value), other values: "average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param dist: distance ("euclidean", default value), "euclidean", "maximum", "manhattan", "canberra", "binary" "minkowski"
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie","Genre", "Date.d.analyse","Origine","Ruche", "Nutrition" , "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param nc: number of clusters (nc=4, default value)
@param dendrogram: dendrogram and factor map
@return output figures and statistics
@examples Lt<-Vizclus(df_Peaks,varCat1="Taxonomie", value="Pediococcus pentosaceus"),
Lt<-Vizclus(df_Peaks,varCat1="Ruche", value="VI")
Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="Caraway", nc=2),
Lt<-Vizclus(df_Peaks,varCat1="Genre", value="Lactobacillus", nc=4)
Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="All"),
Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="Caraway", nc=3, graph="fm")

source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust
https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz
http://www.sthda.com/english/wiki/beautiful-dendrogram-visualizations-in-r-5-must-known-methods-unsupervised-machine-learning

Lt<-Vizclus(df_Peaks,varCat1="Genre", value="Lactobacillus", nc=4)

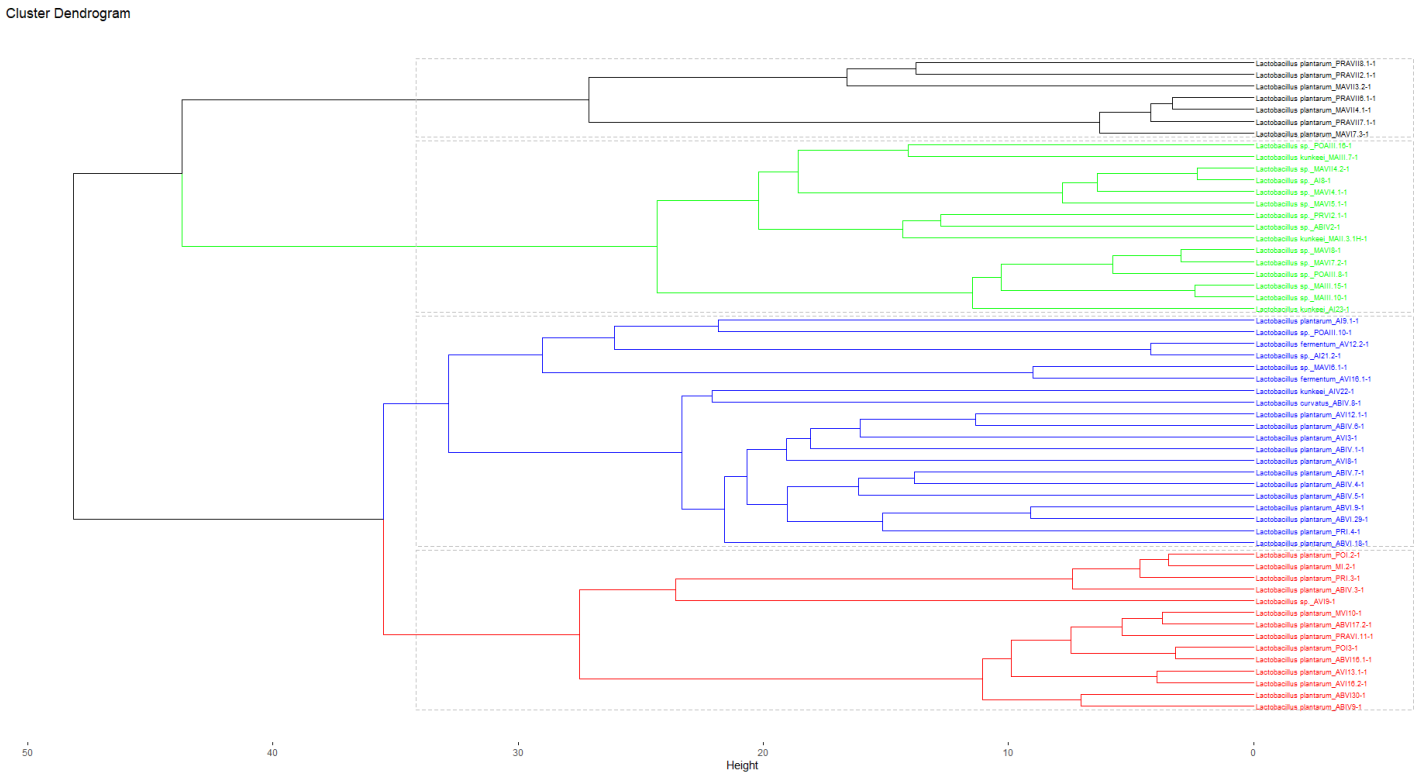
Output:

```
[1] "clustering vector: cluster assignment to each isolate"
  Lactobacillus plantarum_ABIV.1-1   Lactobacillus plantarum_ABIV.3-1   Lactobacillus plantarum_ABIV.4-1
                                 1                                 2                                 1
  Lactobacillus plantarum_ABIV.5-1   Lactobacillus plantarum_ABIV.6-1   Lactobacillus plantarum_ABIV.7-1
                                 1                                 1                                 1
   Lactobacillus curvatus_ABIV.8-1  Lactobacillus plantarum_ABVI.18-1  Lactobacillus plantarum_ABVI.29-1
                                 1                                 1                                 1
  Lactobacillus plantarum_ABVI.9-1  Lactobacillus kunkeei_MAII.3.1H-1      Lactobacillus sp._MAIII.10-1
                                 1                                 3                                 3
     Lactobacillus sp._MAIII.15-1    Lactobacillus kunkeei_MAIII.7-1     Lactobacillus plantarum_MI.2-1
                                 3                                 3                                 2
     Lactobacillus sp._POAIII.10-1     Lactobacillus sp._POAIII.16-1      Lactobacillus sp._POAIII.8-1
                                 1                                 3                                 3
   Lactobacillus plantarum_POI.2-1 Lactobacillus plantarum_PRAVI.11-1    Lactobacillus plantarum_PRI.3-1
                                 2                                 2                                 2
   Lactobacillus plantarum_PRI.4-1       Lactobacillus sp._AI21.2-1   Lactobacillus fermentum_AV12.2-1
                                 1                                 1                                 1
 Lactobacillus plantarum_AVI12.1-1  Lactobacillus fermentum_AVI16.1-1  Lactobacillus plantarum_AVI16.2-1
                                 1                                 1                                 2
    Lactobacillus plantarum_AVI3-1     Lactobacillus plantarum_AVI8-1         Lactobacillus sp._AVI9-1
                                 1                                 1                                 2
```

……..shortened output……..

```
 [1] "The size of each cluster"
  cl
 1  2  3  4
20 14 15  7
[1] "Cophenetic  coeficient"
```

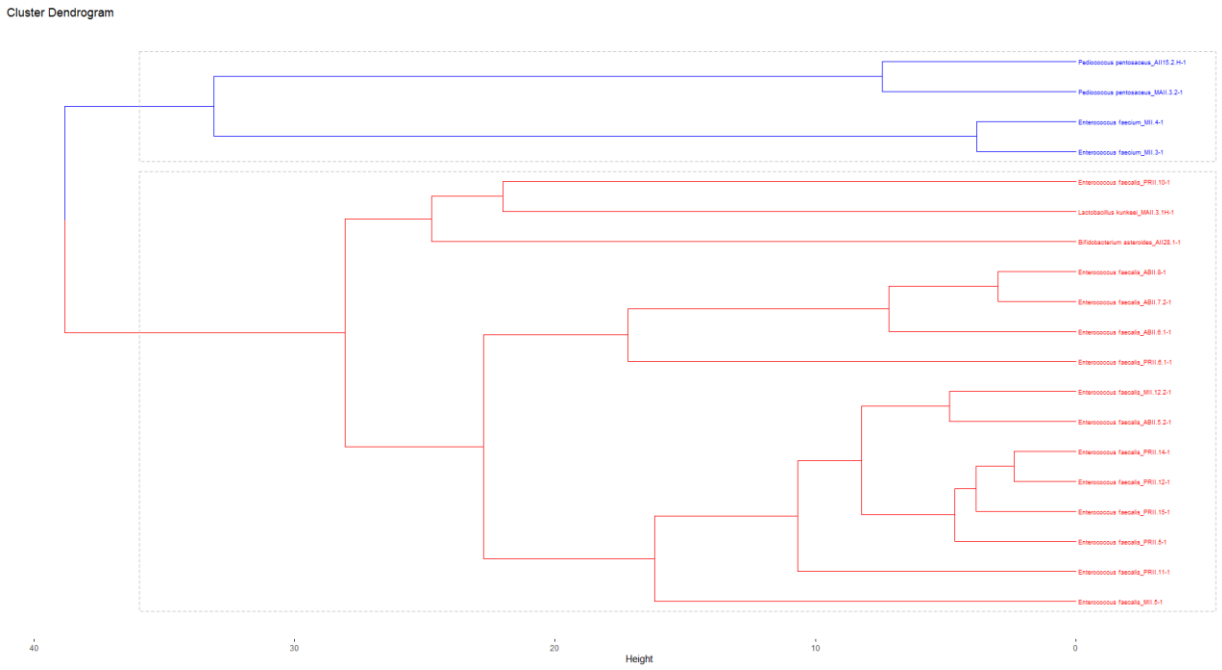
[1] 0.6082436



Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="Caraway", nc=2)

Output:

```
[1] "clustering vector: cluster assignment to each isolate"
  Enterococcus faecalis_ABII.5.2-1   Enterococcus faecalis_ABII.6.1-1   Enterococcus faecalis_ABII.7.2-1
                                 1                                 1                                 1
    Enterococcus faecalis_ABII.8-1   Lactobacillus kunkeei_MAII.3.1H-1  Pediococcus pentosaceus_MAII.3.2-1
                                 1                                 1                                 2
  Enterococcus faecalis_MII.12.2-1      Enterococcus faecium_MII.3-1      Enterococcus faecium_MII.4-1
                                 1                                 2                                 2
     Enterococcus faecalis_MII.5-1    Enterococcus faecalis_PRII.10-1    Enterococcus faecalis_PRII.11-1
                                 1                                 1                                 1
   Enterococcus faecalis_PRII.12-1    Enterococcus faecalis_PRII.14-1    Enterococcus faecalis_PRII.15-1
                                 1                                 1                                 1
    Enterococcus faecalis_PRII.5-1   Enterococcus faecalis_PRII.6.1-1 Pediococcus pentosaceus_AII15.2.H-1
                                 1                                 1                                 2
Bifidobacterium asteroides_AII28.1-1
                                 1
[1] "The size of each cluster"
cl
 1  2
15  4
[1] "Cophenetic coeficient"
[1] 0.9115426
```



Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="Caraway", nc=3,graph="fm")

[1] "clustering vector: cluster assignment to each isolate"

Enterococcus faecalis_ABII.5.2-1	Enterococcus faecalis_ABII.6.1-1	Enterococcus faecalis_ABII.7.2-1
1	1	1
Enterococcus faecalis_ABII.8-1	Lactobacillus kunkeei_MAII.3.1H-1	Pediococcus pentosaceus_MAII.3.2-1
1	1	2
Enterococcus faecalis_MII.12.2-1	Enterococcus faecium_MII.3-1	Enterococcus faecium_MII.4-1
1	3	3
Enterococcus faecalis_MII.5-1	Enterococcus faecalis_PRII.10-1	Enterococcus faecalis_PRII.11-1
1	1	1
Enterococcus faecalis_PRII.12-1	Enterococcus faecalis_PRII.14-1	Enterococcus faecalis_PRII.15-1
1	1	1
Enterococcus faecalis_PRII.5-1	Enterococcus faecalis_PRII.6.1-1	Pediococcus pentosaceus_AII15.2.H-1
1	1	2

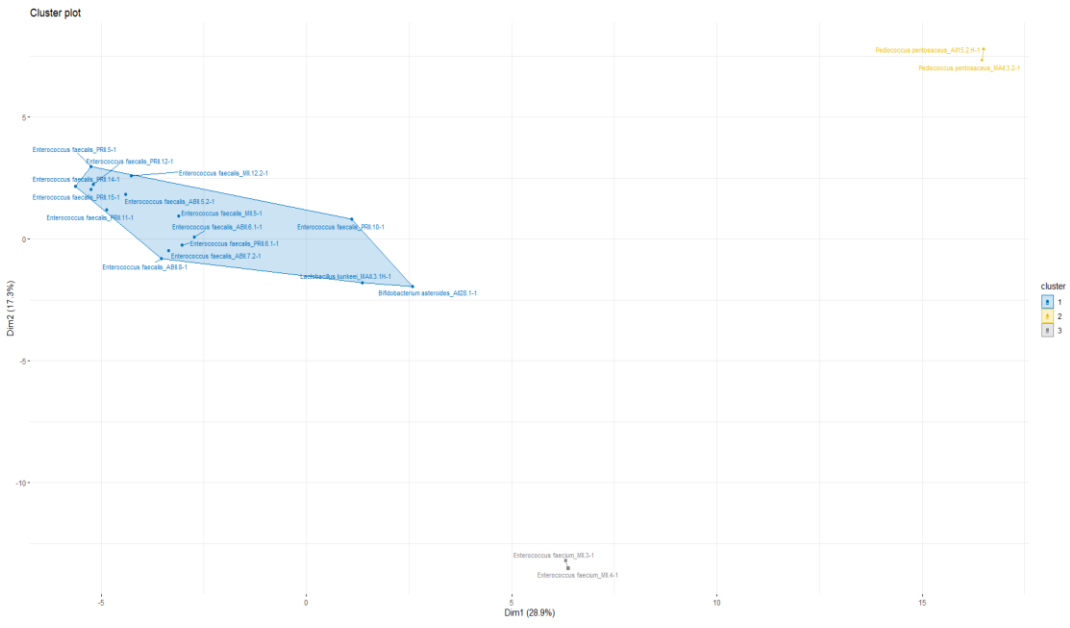
Bifidobacterium asteroides_AII28.1-1
1

[1] "The size of each cluster"

cl
1 2 3
15 2 2

[1] "Cophenetic coeficient"

[1] 0.9115426



Lt<-Vizclus(df_Peaks,varCat1="Nutrition", value="All")

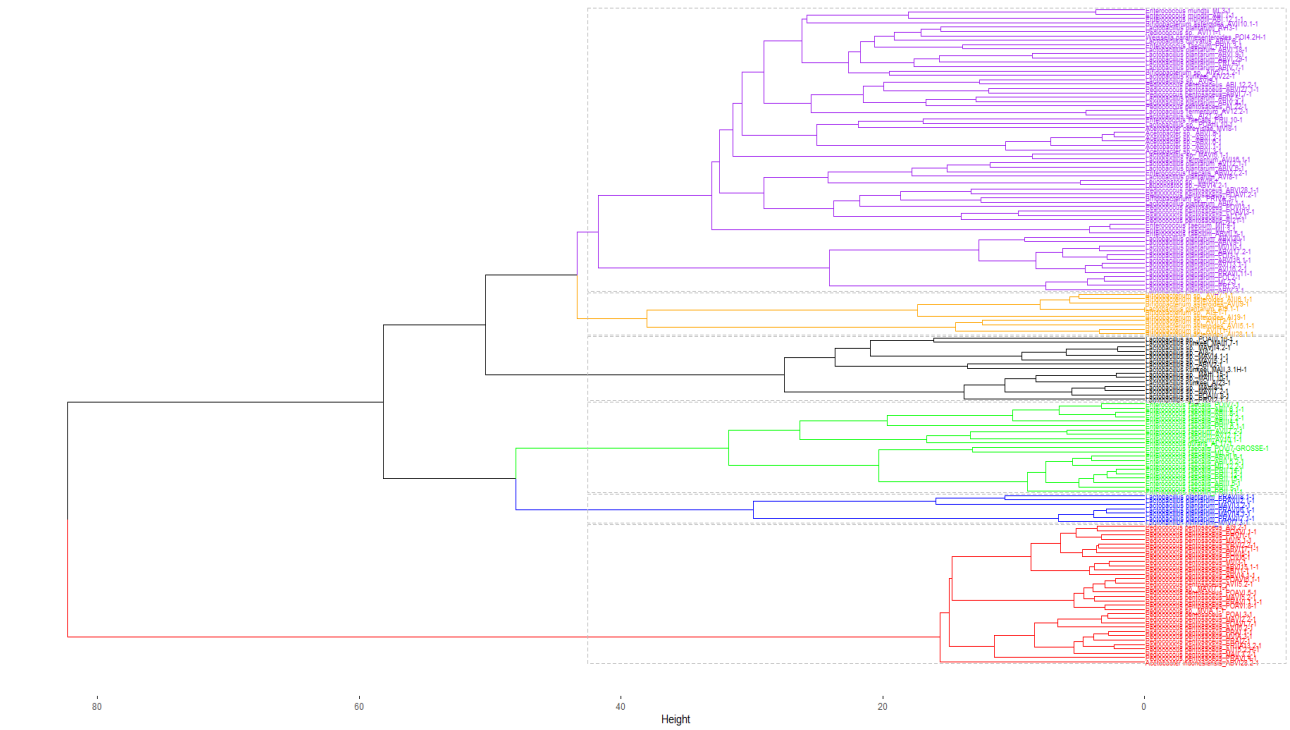
Output:

```
[1] "clustering vector: cluster assignment to each isolate"
    Pediococcus pentosaceus_AI.12-1        Enterococcus durans_AI.17-1     Pediococcus pentosaceus_AI.22-1
                                  1                                  2                                  1
     Enterococcus mundtii_ABI.12.1-1   Pediococcus pentosaceus_ABI.12.2-1        Enterococcus mundtii_ABI.7-1
                                  1                                  1                                  1
    Enterococcus faecalis_ABII.5.2-1    Enterococcus faecalis_ABII.6.1-1    Enterococcus faecalis_ABII.7.2-1
                                  2                                  2                                  2
     Enterococcus faecalis_ABII.8-1      Enterococcus faecalis_ABIII.5-1    Lactobacillus plantarum_ABIV.1-1
                                  2                                  2                                  1
    Lactobacillus plantarum_ABIV.3-1    Lactobacillus plantarum_ABIV.4-1    Lactobacillus plantarum_ABIV.5-1
                                  1                                  1                                  1
    Lactobacillus plantarum_ABIV.6-1    Lactobacillus plantarum_ABIV.7-1     Lactobacillus curvatus_ABIV.8-1
                                  1                                  1                                  1
         Acetobacter sp._ABVI.1-1    Lactobacillus plantarum_ABVI.18-1          Acetobacter sp._ABVI.2-1
                                  1                                  1                                  1
   Lactobacillus plantarum_ABVI.29-1          Acetobacter sp._ABVI.3-1          Acetobacter sp._ABVI.6-1
                                  1                                  1                                  1
    Pediococcus pentosaceus_ABVI.7-1          Acetobacter sp._ABVI.8-1    Lactobacillus plantarum_ABVI.9-1
                                  1                                  1                                  1
     Enterococcus faecium_ABVII.5-1     Enterococcus faecalis_ABVII.6-1    Lactobacillus kunkeei_MAII.3.1H-1
                                  1                                  2                                  3
  Pediococcus pentosaceus_MAII.3.2-1       Lactobacillus sp._MAIII.10-1       Lactobacillus sp._MAIII.15-1
                                  4                                  3                                  3
     Lactobacillus kunkeei_MAIII.7-1     Lactobacillus plantarum_MI.2-1         Enterococcus mundtii_MI.3-1
                                  3                                  1                                  1
    Enterococcus faecalis_MII.12.2-1       Enterococcus faecium_MII.3-1       Enterococcus faecium_MII.4-1
                                  2                                  1                                  1
      Enterococcus faecalis_MII.5-1          Pediococcus sp._MVIA.1-1    Pediococcus pentosaceus_MVIA.3-1
```

.....shortened output.....

```
[1] "The size of each cluster"
cl
 1  2  3  4
75 28 15 32
[1] "Cophenetic  coeficient"
[1] 0.39633
```

Cluster Dendrogram



3.5. Dendrogram_pairComp

Dendrogram_pairComp computes pairwise dendrogram alignments and correlations

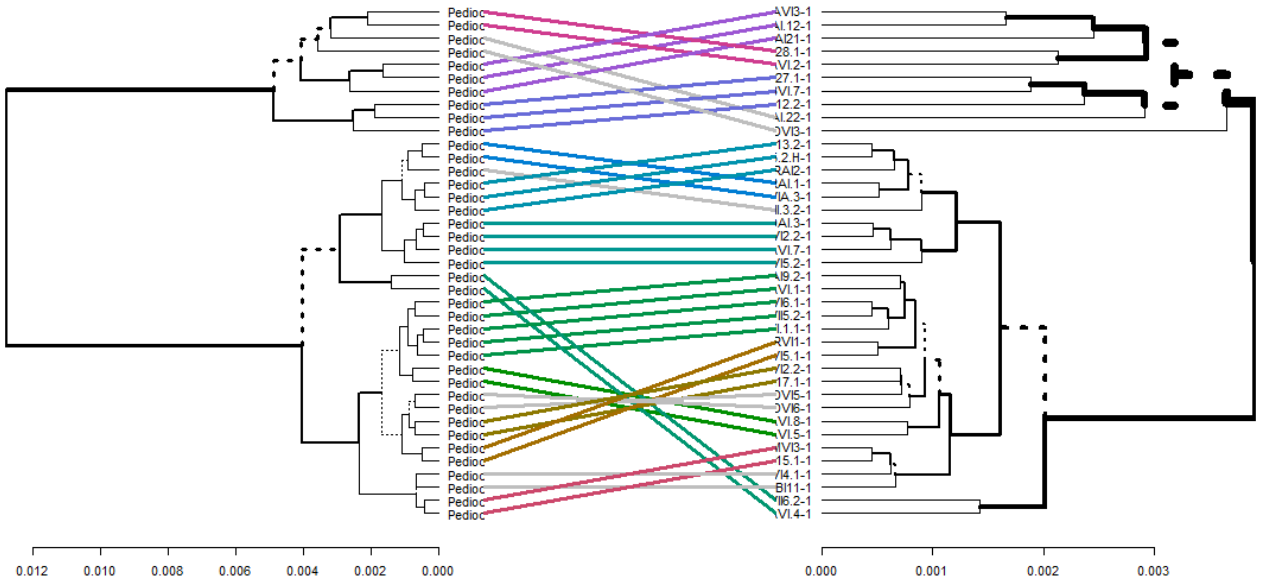
@param df_m: dataframe containing peaks and metadata
@param methCorr: dendrogram correlation method ("cophenetic", default), others: "baker", "common_nodes", "FM_index"
@param meth1, meth2...meth6: hierarchical clustering algorithms "ward2" ,"average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param dist1, dist2, ...dist6: distances "euclidean", "maximum", "manhattan", "canberra", "binary" "minkowski"
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie", "Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: value of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("P ediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@return untangled and tangled dendrograms, dendrogram correlation matrix and statistics
@examples mc<-Dendrogram_pairComp(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus"), mc<-Dendrogram_pairComp(df_Peaks, varCat1="Nutrition", value="Marrubium vulgare")

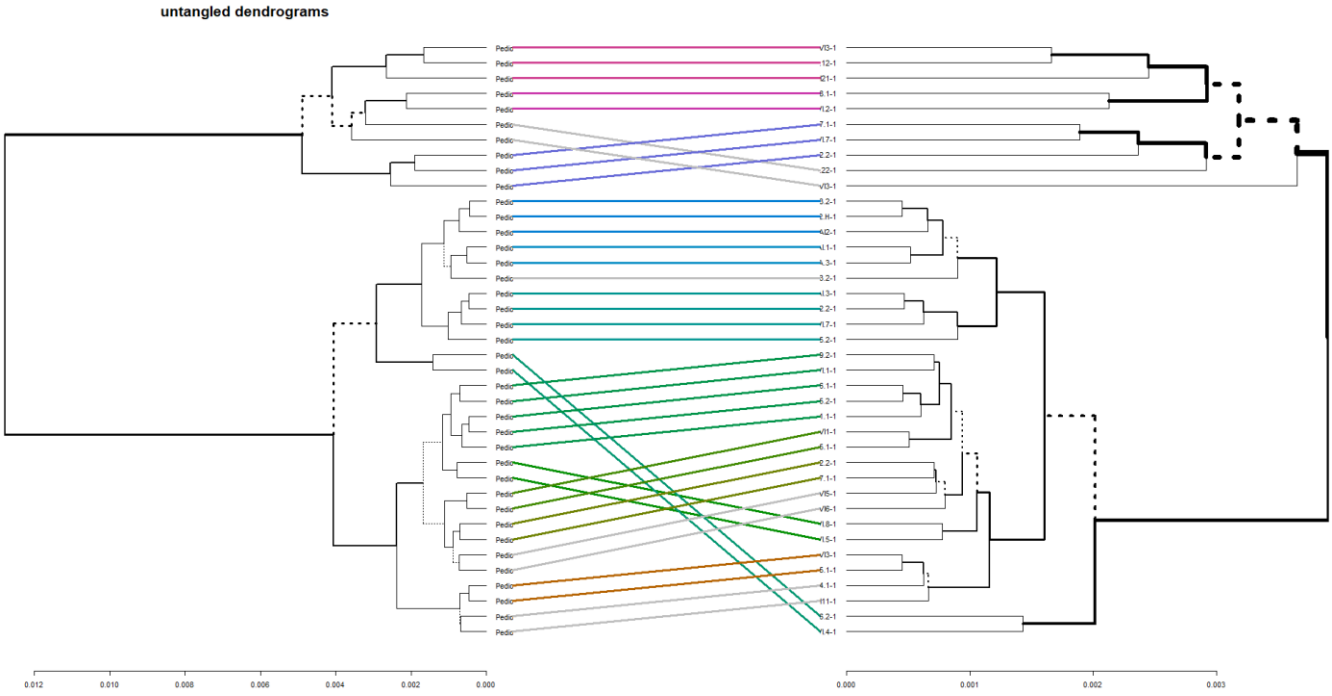
source: https://cran.r-project.org/web/packages/dendextend/vignettes/dendextend.html#:~:text=The%20dendextend%20package%20offers%20a,its%20branches%2C%20nodes%20and%20labels.
https://www.rdocumentation.org/packages/dendextend/versions/1.14.0
https://www.r-graph-gallery.com/340-custom-your-dendrogram-with-dendextend.html
https://academic.oup.com/bioinformatics/article/31/22/3718/240978
https://cran.rstudio.com/web/packages/dendextend/vignettes/Cluster_Analysis.html
https://www.datanovia.com/en/lessons/comparing-cluster-dendrograms-in-r/
https://rdrr.io/cran/dendextend/man/untangle.html

mc<-Dendrogram_pairComp(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus")

output:

```
[1] "Alignment score"
[1] 0.06543847
[1] "Cophenetic correlation coefficient"
[1] 0.9561887
[1] "Baker correlation coefficient"
[1] 0.9671598
[1] "n1: ward.D2 euclidean"
[1] "n2: mcquitty euclidean"
[1] "n3: complete euclidean"
[1] "n4: centroid euclidean"
[1] "n5: single euclidean"
[1] "n6: average euclidean"
[1] "Tree correlation matrix"
   n1   n2   n3   n4   n5   n6
n1 1.00 0.96 0.95 0.87 0.91 0.96
n2 0.96 1.00 0.98 0.95 0.98 1.00
n3 0.95 0.98 1.00 0.93 0.97 0.98
n4 0.87 0.95 0.93 1.00 0.97 0.95
n5 0.91 0.98 0.97 0.97 1.00 0.99
n6 0.96 1.00 0.98 0.95 0.99 1.00
```

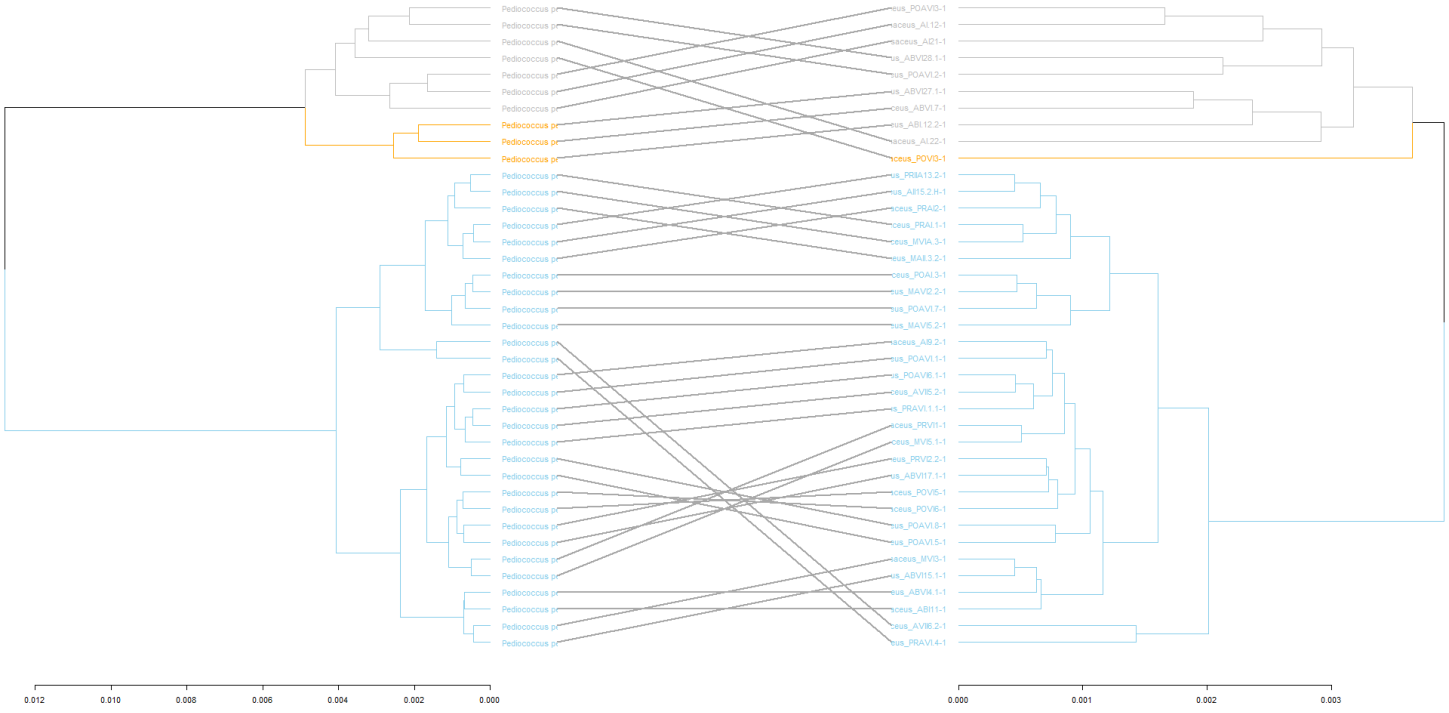

0.012
0.010
0.008
0.006
0.004
0.002
0.000
0.000
0.001
0.002
0.003

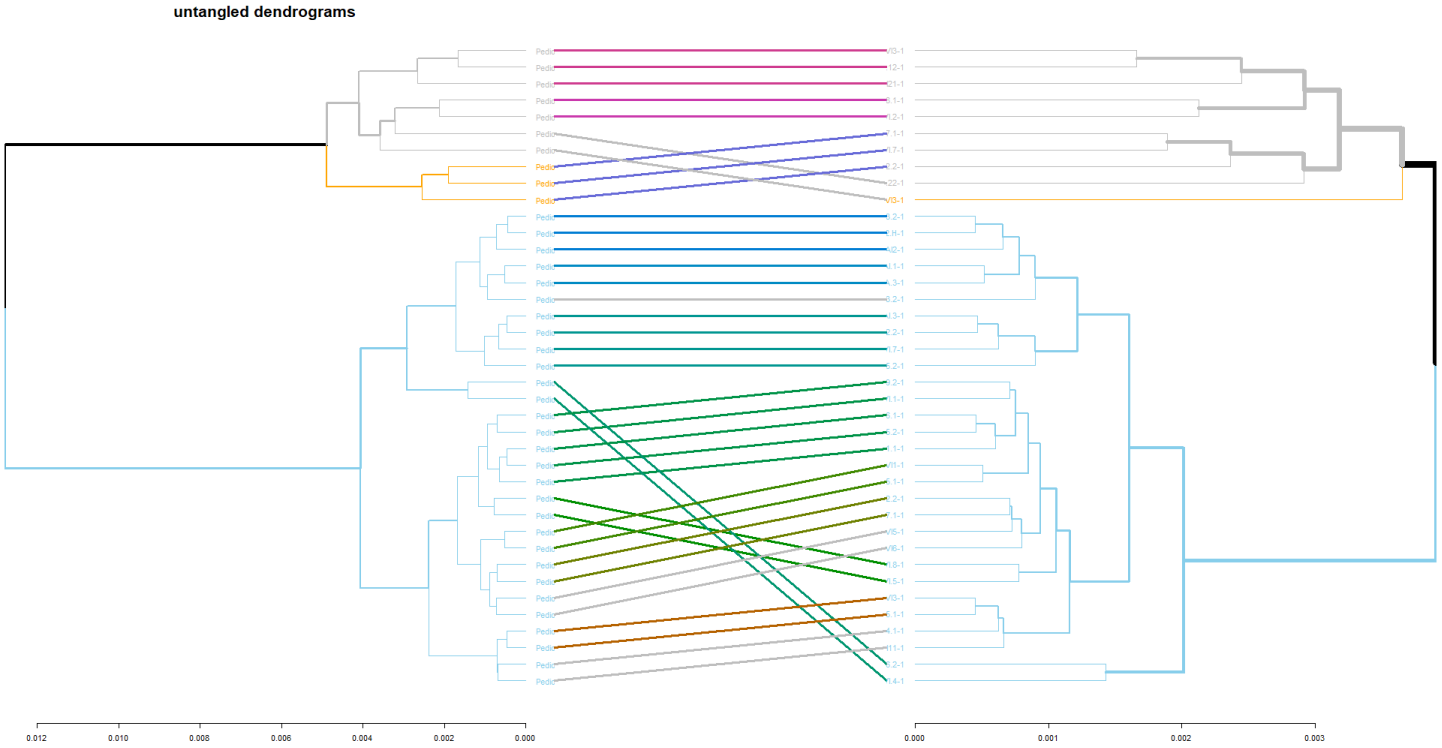

untangled dendrograms

mc<-Dendrogram_pairComp(df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", graph="ClusterDends")

[1] "Alignment score"
[1] 0.06543847
[1] "Cophenetic correlation coefficient"
[1] 0.9561887
[1] "Baker correlation coefficient"
[1] 0.9671598
[1] "n1: ward.D2 euclidean"
[1] "n2: mcquitty euclidean"
[1] "n3: complete euclidean"
[1] "n4: centroid euclidean"
[1] "n5: single euclidean"
[1] "n6: average euclidean"
[1] "Tree correlation matrix"

	n1	n2	n3	n4	n5	n6
n1	1.00	0.96	0.95	0.87	0.91	0.96
n2	0.96	1.00	0.98	0.95	0.98	1.00
n3	0.95	0.98	1.00	0.93	0.97	0.98
n4	0.87	0.95	0.93	1.00	0.97	0.95
n5	0.91	0.98	0.97	0.97	1.00	0.99
n6	0.96	1.00	0.98	0.95	0.99	1.00





3.6. OptClusters

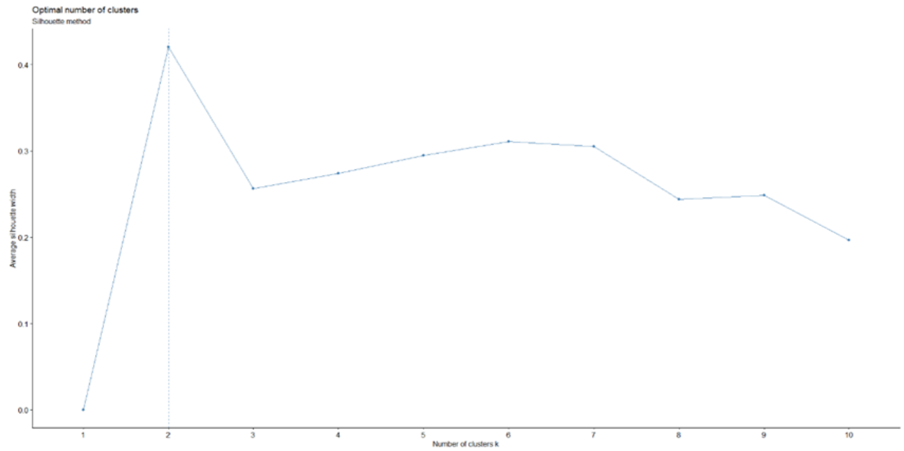
OptClusters, is a wrapper of several functions to visualize and compute optimal clusters for different clustering and evaluation methods

@param df_m: dataframe containing peaks and metadata
@param meth: clustering algorithms "kmeans", (default value), other values: "pam" or "hclust"
@param dist: distances "euclidean", (default value), "maximum", "manhattan", "canberra", "binary" "minkowski"
@param varCat1: categorical variable for choosing isolates, examples : "Taxonomie","Genre", "Date.d.analyse" ,"Origine","Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param minc: minimal number of clusters (minc=2, default value)
@param maxc: maximal number of clusters (maxc=10, default value)
@param ind: methods to evaluate clustering algorithms: "total within sum of squares" ,"average silhouette width" and "gap statistics"
@param nb: number of bootstrap samples (nb=100, default value)
@return figures and statistics
@examples OptClusters(df_Peaks, varCat1="Taxonomie", value="Enterococcus faecalis"),
OptClusters(df_Peaks, varCat1="Taxonomie" , value="All")
OptClusters(df_Peaks, meth="pam", varCat1="Taxonomie", value="All", ind="gap statistics"),
OptClusters(df_Peaks, meth="hclust", varCat1="Taxonomie", value="All", ind="gap statistics")

source: https://rpubs.com/pg2000in/OptimumClusters
http://rstudio-pubs-static.s3.amazonaws.com/265632_3ad9e0b981244e15887677f8dffb39a0.html#using-30-different-indices
https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/
https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_nbclust

OptClusters(df_Peaks, varCat1="Taxonomie”, value="Enterococcus faecalis")

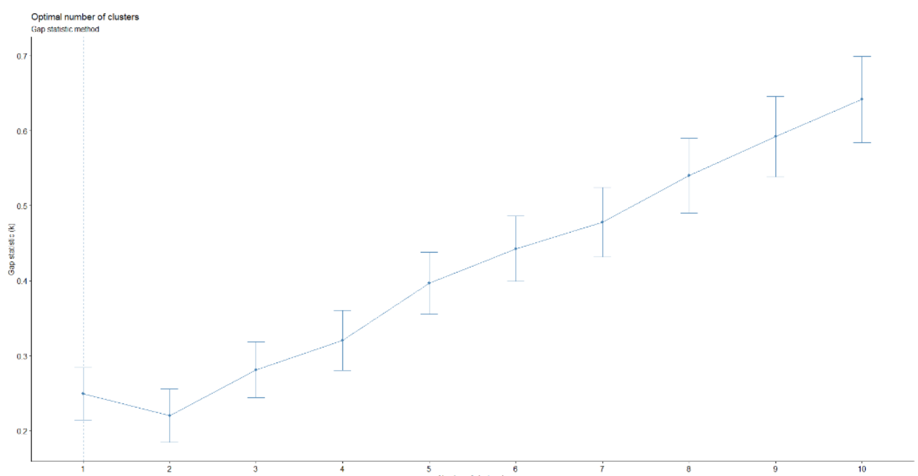
Output:



OptClusters(df_Peaks, varCat1="Taxonomie", value="Enterococcus faecalis", ind="gap statistics")

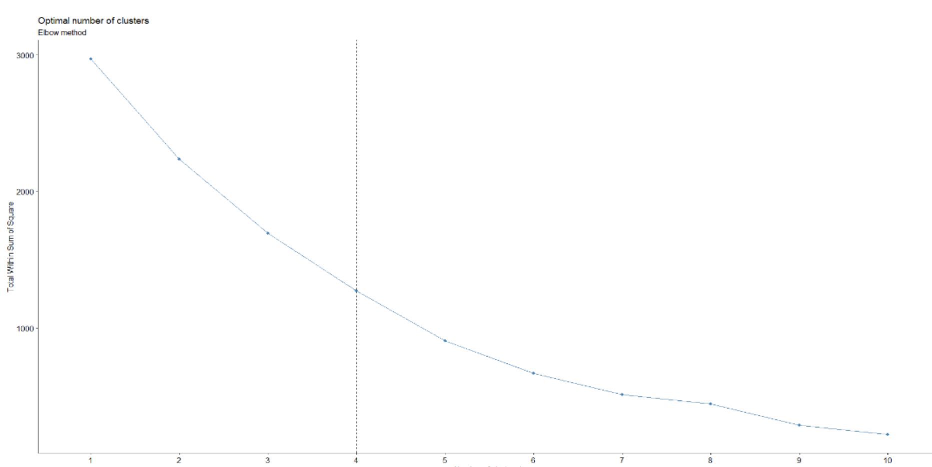
Output:

Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50) [one "." per sample]:
.... 50



OptClusters (df_Peaks, varCat1="Taxonomie", value="Enterococcus faecalis", ind="total within sum of squares")

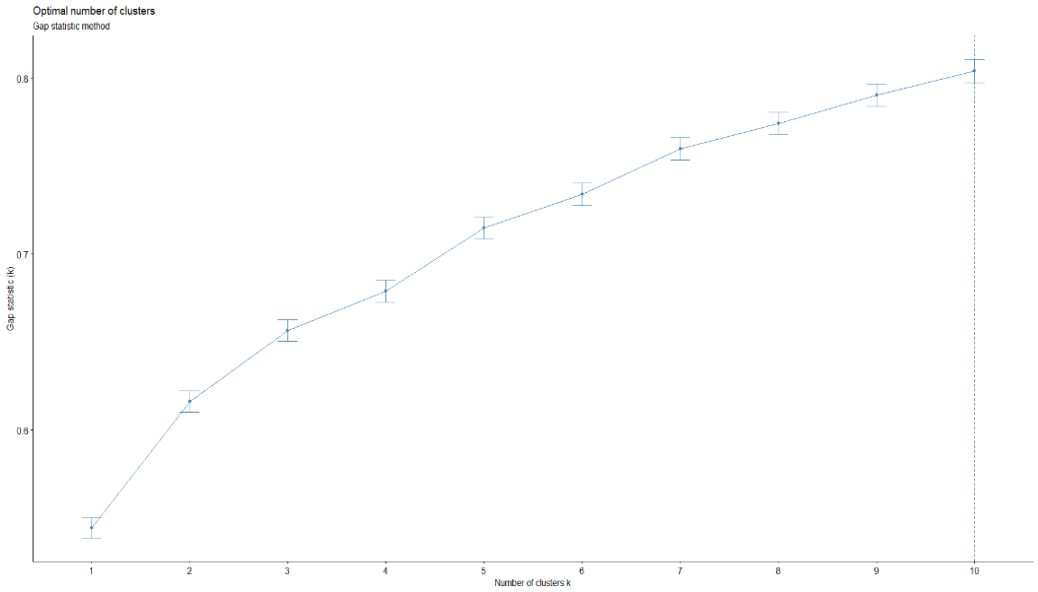
Output:



OptClusters(df_Peaks, meth="pam", varCat1="Taxonomie", value="All", ind="gap statistics", nb=500)

Output:

Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 500) [one "." per sample]:
..... 50
..... 100
..... 150
..... 200
..... 250
..... 300
..... 350
..... 400
..... 450
..... 500



3.7. VisualOptClusters

visualization and statistics for MALDI_TOF spectra clustering validation

@param df_m: dataframe containing peaks and metadata
@param meth: clustering algorithms "hclust", (default value), other values: "kmeans", "pam", "clara", "fanny", "hclust","agnes", "diana"
@param dist: distances "euclidean", (default value), "maximum", "manhattan", "canberra", "binary" "minkowski"
@param meth2: hc methods "ward.D2", (default value), average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param varCat1: categorical variable for choosing isolates, examples: “Taxonomie", "Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param nc: number of clusters, nc=3, (default value)
@return figures and statistics
@examples s<-VisualOptClusters(df_Peaks, varCat1="Genre", value="Enterococcus"),
s<-VisualOptClusters(df_Peaks, meth="pam", varCat1="Genre", value="Lactobacillus", nc=2)
s<-VisualOptClusters(df_Peaks, meth="hclust", meth2="average", dist="pearson", varCat1="Genre", value="All", nc=5),
s<-VisualOptClusters(df_Peaks, meth="kmeans", dist="euclidean", varCat1="Genre",

value="All", nc=3)

source: https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_cluster
https://afit-r.github.io/kmeans_clustering
http://rstudio-pubs-static.s3.amazonaws.com/265632_3ad9e0b981244e15887677f8dffb39a0.html#using-30-different-indices

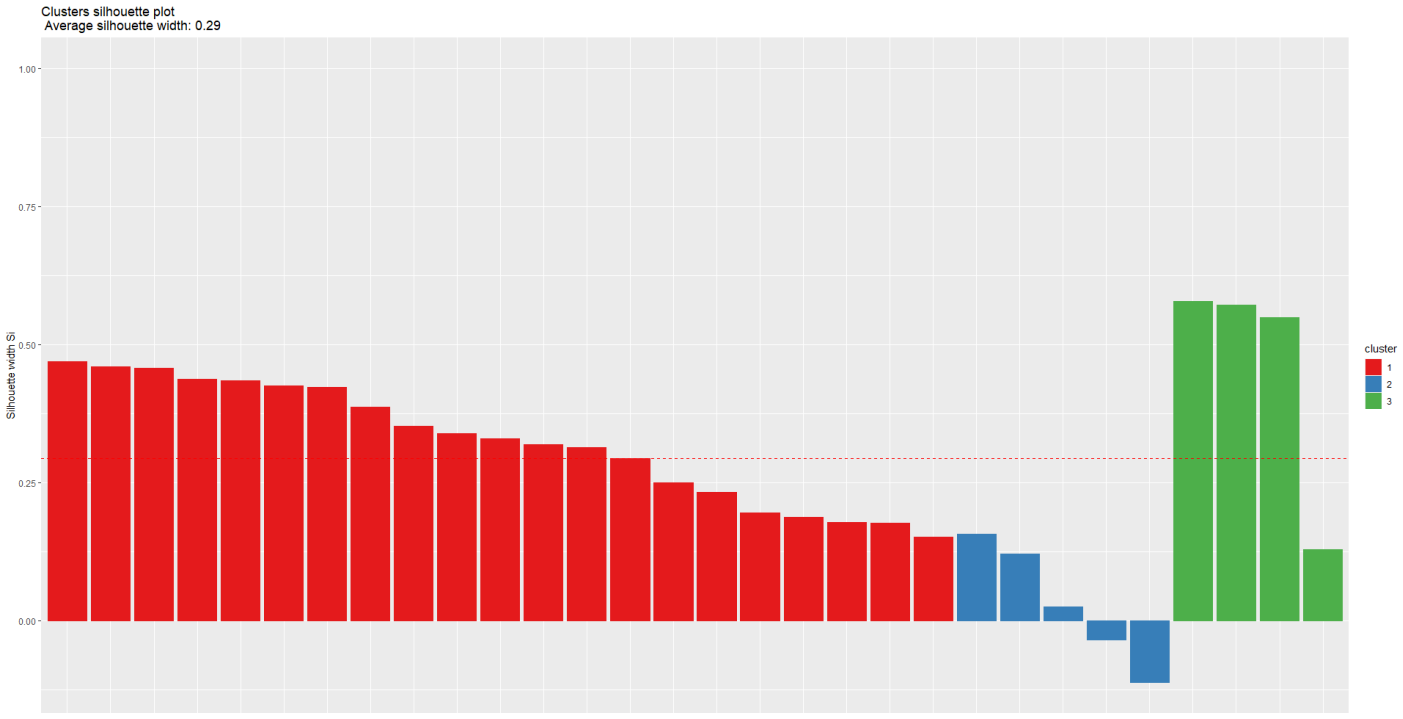
s<-VisualOptClusters(df_Peaks, varCat1="Genre", value="Enterococcus")

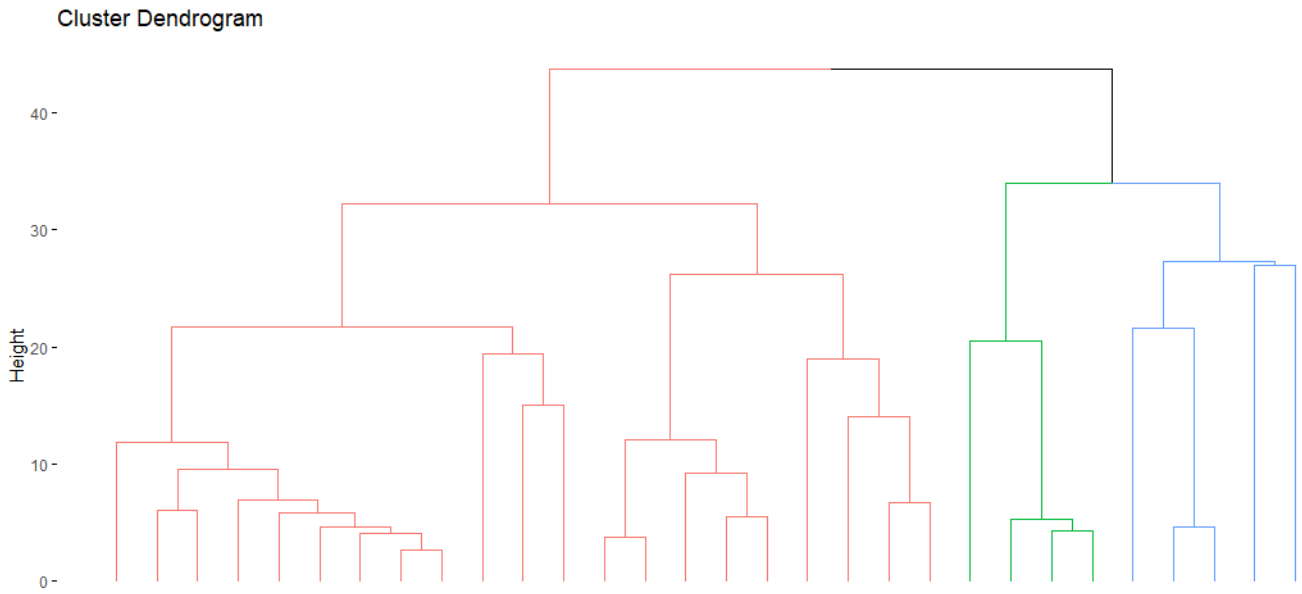
output:

```
  cluster size ave.sil.width
1       1   21         0.32
2       2    5         0.03
3       3    4         0.46
$widths
                                cluster neighbor    sil_width
Enterococcus faecalis_ABIII.5-1       1        3  0.46934828
Enterococcus faecalis_PRII.14-1       1        3  0.46005835
Enterococcus faecalis_PRII.12-1       1        3  0.45779439
Enterococcus faecalis_PRII.5-1        1        3  0.43660605
Enterococcus faecalis_PRII.15-1       1        3  0.43493818
Enterococcus faecalis_ABVII.6-1       1        3  0.42491011
Enterococcus faecalis_ABII.5.2-1      1        3  0.42264275
Enterococcus faecalis_MII.12.2-1      1        3  0.38646783
Enterococcus faecalis_PRII.11-1       1        3  0.35187800
Enterococcus faecalis_ABII.6.1-1      1        3  0.33850791
Enterococcus faecalis_ABII.7.2-1      1        3  0.32897691
Enterococcus faecalis_ABII.8-1        1        3  0.31944142
Enterococcus faecalis_POIV7-1         1        3  0.31295360
Enterococcus faecalis_ABIII4-1        1        3  0.29358295
Enterococcus faecalis_MII.5-1         1        3  0.24999632
Enterococcus faecalis_PRII.6.1-1      1        3  0.23297577
Enterococcus faecium_AVII7.2-1        1        3  0.19485022
Enterococcus faecium_AV10.1-1         1        3  0.18642428
Enterococcus faecium_AV12.1-1         1        3  0.17833360
```

.....shortened output.....

```
$clus.avg.widths
[1] 0.32418331 0.03065726 0.45669725

$avg.width
[1] 0.2929308
```





s<-VisualOptClusters(df_Peaks, meth="pam",varCat1="Genre", value="All", nc=6)

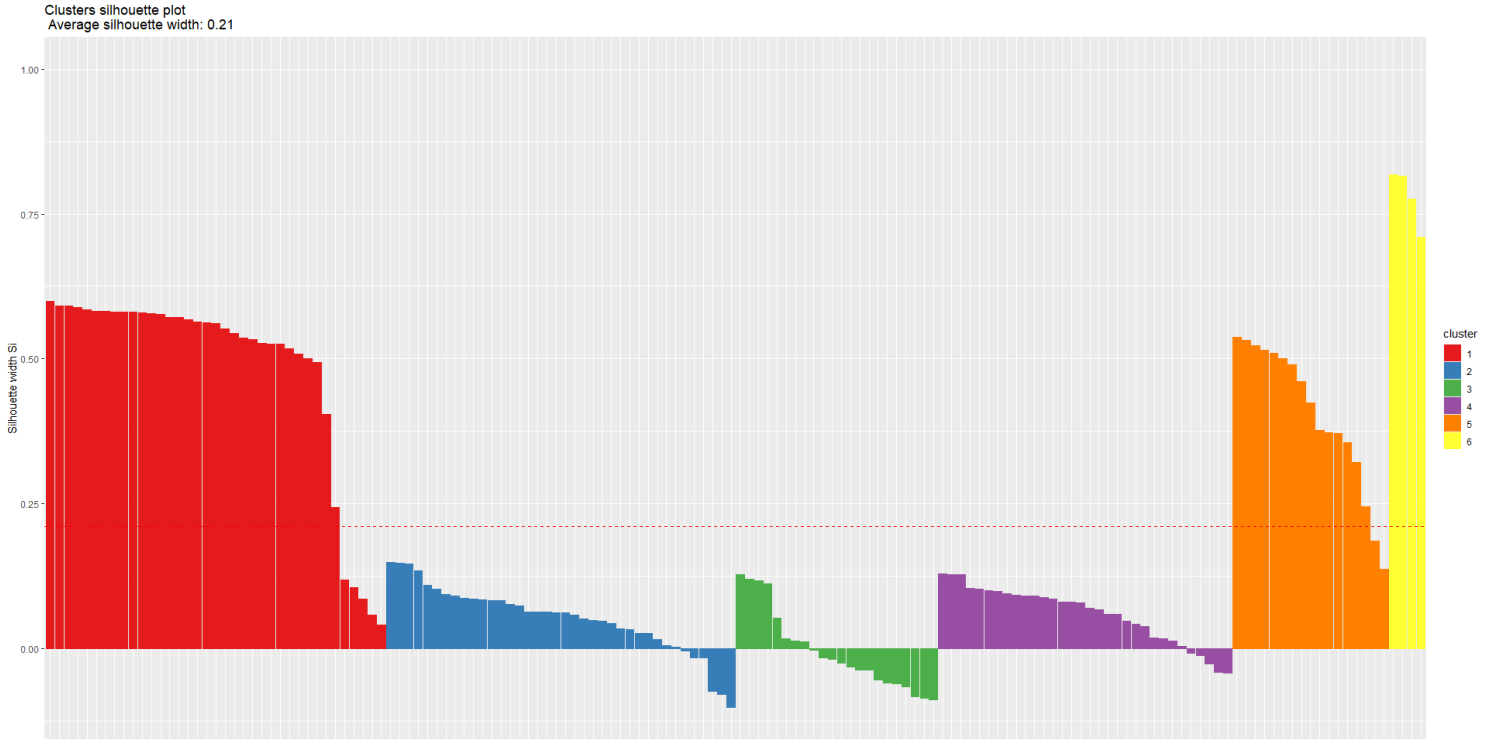
output:

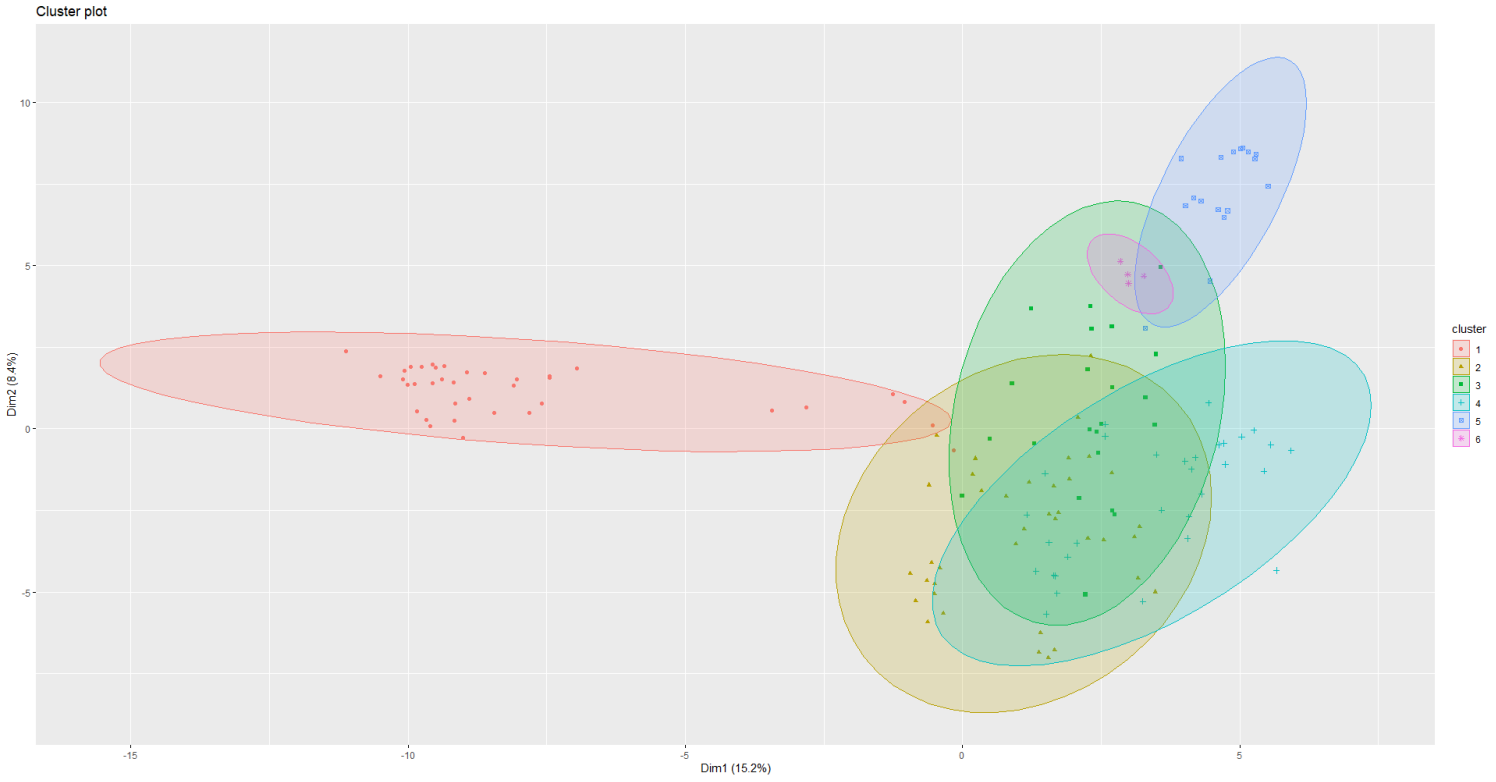
```
  cluster  size ave.sil.width
1       1  37          0.48
2       2  38          0.05
3       3  22          0.00
4       4  32          0.06
5       5  17          0.40
6       6   4          0.78
$widths
                                  cluster  neighbor    sil_width
Pediococcus pentosaceus_POAVI.1-1       1       2  0.599407924
Pediococcus pentosaceus_PRAVI.1.1-1     1       2  0.591676671
Pediococcus pentosaceus_PRAI.1-1        1       2  0.591085652
Pediococcus pentosaceus_AVII5.2-1       1       2  0.588123484
Pediococcus pentosaceus_PRVI2.2-1       1       2  0.584962734
Pediococcus pentosaceus_POAVI.7-1       1       2  0.582671602
Pediococcus pentosaceus_PRIIA13.2-1     1       2  0.582162559
Pediococcus pentosaceus_MVIA.3-1        1       2  0.580898030
Pediococcus pentosaceus_POAVI.5-1       1       2  0.580663177
Pediococcus pentosaceus_AI9.2-1         1       2  0.580299720
Pediococcus pentosaceus_PRVI1-1         1       2  0.579086415
Pediococcus pentosaceus_MAVI2.2-1       1       2  0.577626216
Pediococcus pentosaceus_MVI5.1-1        1       2  0.577150065
Pediococcus pentosaceus_POVI5-1         1       2  0.572156201
Pediococcus pentosaceus_ABVI15.1-1      1       2  0.571275241
Pediococcus pentosaceus_AII15.2.H-1     1       2  0.567036446
Pediococcus pentosaceus_POAVI6.1-1      1       2  0.564319053
Pediococcus pentosaceus_POAI.3-1        1       2  0.562481769
Pediococcus pentosaceus_ABVI17.1-1      1       2  0.560626245
Pediococcus pentosaceus_PRAI2-1         1       2  0.552265722
Pediococcus pentosaceus_MVI3-1          1       2  0.544002379
Pediococcus pentosaceus_POVI6-1         1       2  0.535424547
Pediococcus pentosaceus_MAVI5.2-1       1       2  0.533393236
Pediococcus pentosaceus_ABVI4.1-1       1       2  0.526594660
```

……shortened output……

$clus.avg.widths

[1] 0.481352515 0.051162303 -0.004511589 0.058257924 0.403354044 0.779722190





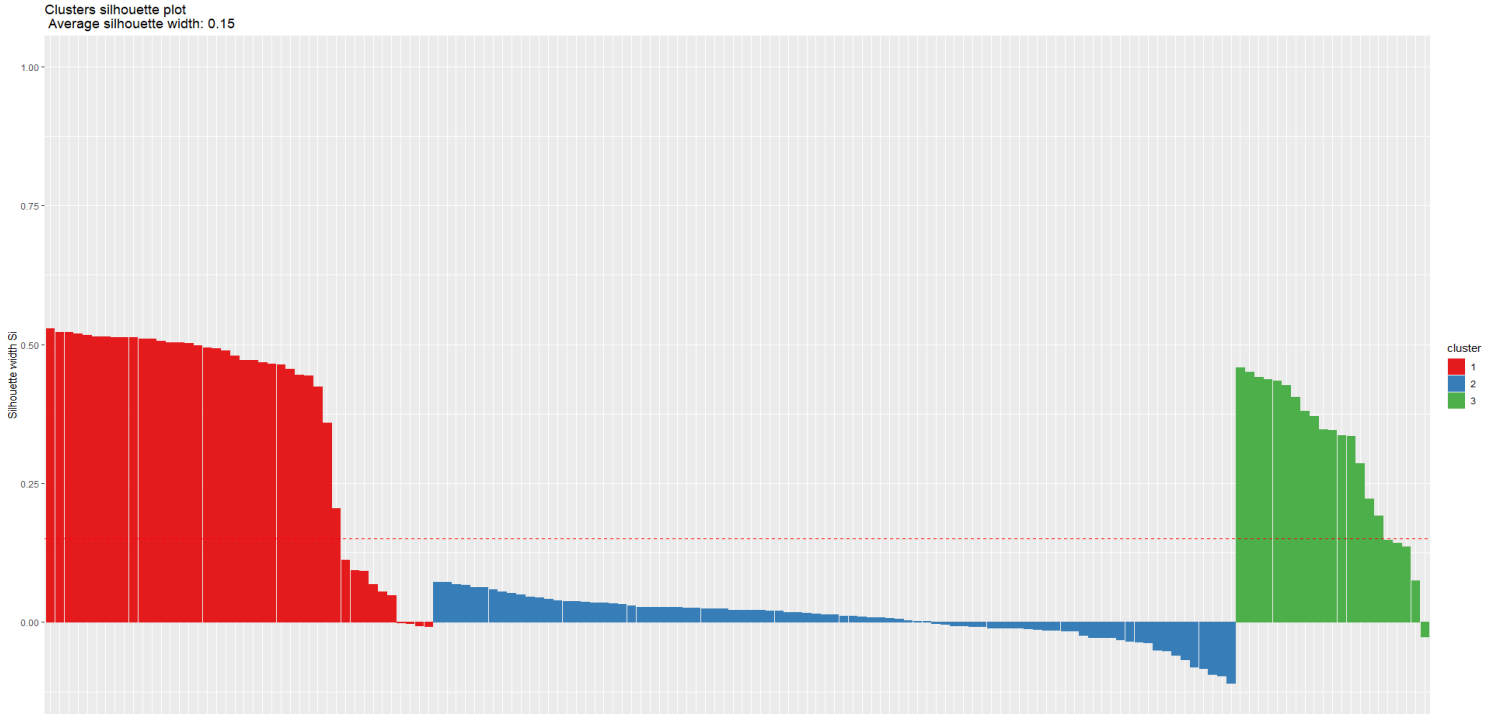
s<-VisualOptClusters(df_Peaks, meth="pam",varCat1="Genre",value="All", nc=3)

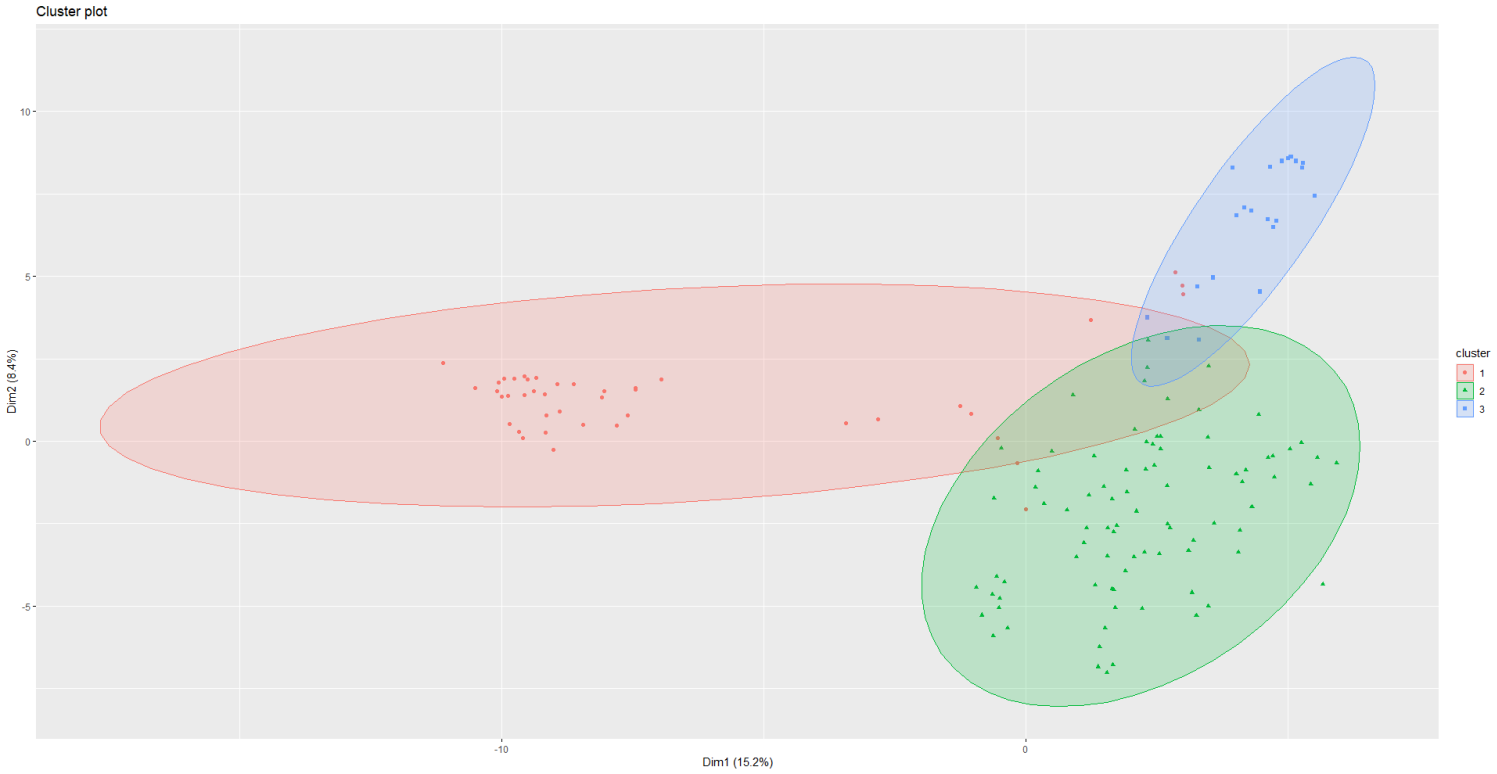
output:

```
  cluster size ave.sil.width
1       1   42         0.38
2       2   87         0.01
3       3   21         0.30
$widths
                                cluster  neighbor     sil_width
Pediococcus pentosaceus_POAVI.1-1       1       2  0.5282162251
Pediococcus pentosaceus_PRAVI.1.1-1     1       2  0.5217488452
Pediococcus pentosaceus_PRAI.1-1        1       2  0.5216367308
Pediococcus pentosaceus_AVII5.2-1       1       3  0.5188998087
Pediococcus pentosaceus_PRVI2.2-1       1       2  0.5170117138
Pediococcus pentosaceus_PRIIA13.2-1     1       2  0.5147840466
Pediococcus pentosaceus_MVIA.3-1        1       2  0.5137295014
Pediococcus pentosaceus_POAVI.7-1       1       2  0.5133880537
Pediococcus pentosaceus_AI9.2-1         1       3  0.5128952319
Pediococcus pentosaceus_POAVI.5-1       1       2  0.5122735714
Pediococcus pentosaceus_PRVI1-1         1       2  0.5106237133
Pediococcus pentosaceus_MVI5.1-1        1       2  0.5105702308
Pediococcus pentosaceus_MAVI2.2-1       1       3  0.5056111596
Pediococcus pentosaceus_ABVI15.1-1      1       2  0.5033879340
Pediococcus pentosaceus_POVI5-1         1       2  0.5031626240
Pediococcus pentosaceus_AII15.2.H-1     1       2  0.5022992880
Pediococcus pentosaceus_POAVI6.1-1      1       3  0.4979635584
Pediococcus pentosaceus_ABVI17.1-1      1       2  0.4941022555
Pediococcus pentosaceus_POAI.3-1        1       3  0.4932899968
Pediococcus pentosaceus_PRAI2-1         1       3  0.4896026566
Pediococcus pentosaceus_MVI3-1          1       2  0.4796409880
Pediococcus pentosaceus_POVI6-1         1       2  0.4720077534
Pediococcus pentosaceus_MAVI5.2-1       1       2  0.4718739371
Pediococcus pentosaceus_MAII.3.2-1      1       2  0.4680102940
Pediococcus pentosaceus_ABVI4.1-1       1       2  0.4649807256
Pediococcus pentosaceus_POAVI.8-1       1       2  0.4636068257
Pediococcus sp._MAVI7.1-1               1       3  0.4560500196
Pediococcus pentosaceus_ABI11-1         1       2  0.4454150444
```

.....shortened output.....

```
$clus.avg.widths
[1] 0.375931531 0.006128141 0.302342829

$avg.width
[1] 0.1511431
```





3.8. PointClusterVal

PointClusterVal is a wrapper of two functions for clustering tendency and validation statistics

@param df_m: dataframe containing peaks and metadata
@param meth: clustering algorithms ("hclust", default value), other values: "kmeans", "pam", "clara", "fanny", "hclust","agnes", "diana"
@param dist: distances,"euclidean", (default value), "maximum", "manhattan", "canberra", "binary" "minkowski"
@param meth2: hc methods "ward.D2", (default value), average", "ward.D", "single", "complete", "mcquitty", "median" or "centroid"
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie", "Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition") ,...
@param varCat2: categorical variable for partitioning the set of isolates chosen by using varCat1
@return figures and statistics
@examples ff<-PointClusterVal (df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")

source: https://www.rdocumentation.org/packages/fpc/versions/2.2-8/topics/cluster.stats
https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/

ff<-PointClusterVal(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")

output:

[1] "Clustering Tendency (Hopkins statistic)"

```
$hopkins_stat
[1] 0.7644649

$plot


Call:
stats::hclust(d = x, method = hc_method)

Cluster method   : ward.D2
Distance         : euclidean
Number of objects: 56

$n
[1] 56

$cluster.number
[1] 7

$cluster.size
[1] 14  5 15  4  9  2  7

$min.cluster.size
[1] 2

$noisen
[1] 0

$diameter
[1] 22.95959 19.32593 19.44587 22.41606 10.22588  9.00450 18.59128

$average.distance
[1] 17.622715 10.696875 12.326323 18.685366  6.802126  9.004500 13.368503

$median.distance
[1] 17.689350  7.063469 12.905385 21.432008  7.213886  9.004500 16.300675

$separation
[1] 11.631167  9.989521 14.237874 15.951624  9.989521 17.598738 15.771988

$average.toother
[1] 18.55982 17.98586 18.48002 20.32399 17.17040 21.28370 19.90793

$separation.matrix
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
[1,]  0.00000 13.665972 14.23787 16.94605 11.631167 19.68021 17.11445
[2,] 13.66597  0.000000 15.38696 17.21943  9.989521 19.42117 18.79356
[3,] 14.23787 15.386955  0.00000 16.79070 15.007128 18.44822 15.77199
[4,] 16.94605 17.219435 16.79070  0.00000 15.951624 17.59874 18.81842
[5,] 11.63117  9.989521 15.00713 15.95162  0.000000 18.47002 17.33686
[6,] 19.68021 19.421169 18.44822 17.59874 18.470018  0.00000 21.06001
[7,] 17.11445 18.793559 15.77199 18.81842 17.336860 21.06001  0.00000

$ave.between.matrix
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
[1,]  0.00000 17.80269 18.18237 20.91226 16.70647 21.59085 20.08206
[2,] 17.80269  0.00000 18.62993 19.48475 13.78734 21.12029 20.61810
[3,] 18.18237 18.62993  0.00000 20.02739 17.03173 21.31666 19.13564
[4,] 20.91226 19.48475 20.02739  0.00000 18.71575 21.78504 22.03279
[5,] 16.70647 13.78734 17.03173 18.71575  0.00000 20.03088 19.11152
[6,] 21.59085 21.12029 21.31666 21.78504 20.03088  0.00000 22.03982
[7,] 20.08206 20.61810 19.13564 22.03279 19.11152 22.03982  0.00000

$average.between
[1] 18.69382
```

$average.within
[1] 13.08297

$n.between
[1] 1270

$n.within
[1] 270

$max.diameter
[1] 22.95959

$min.separation
[1] 9.989521

$within.cluster.ss
[1] 4993.077

$clus.avg.silwidths
        1   2   3   4   5   6   7
-0.05629018 0.24330709 0.27923312 0.01124142 0.50783301 0.55037281 0.29374212

$avg.silwidth
[1] 0.2212388

$g2
NULL

$g3
NULL

$pearsongamma
[1] 0.5329142

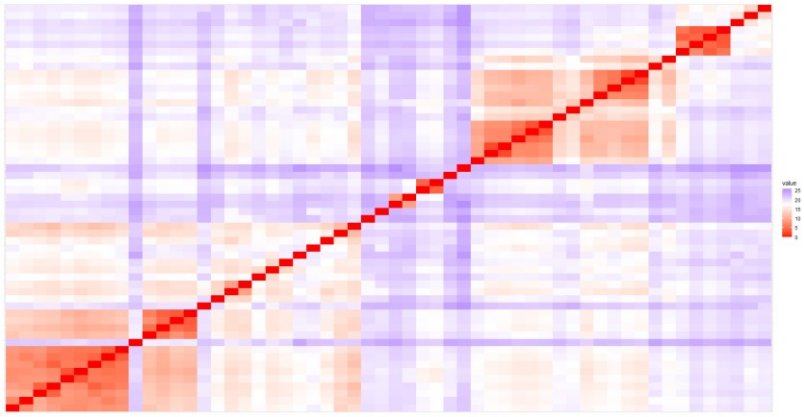
$dunn
[1] 0.4350914

$dunn2
[1] 0.7378681

$entropy
[1] 1.776377

$wb.ratio
[1] 0.6998553

$ch
[1] 6.676386



$cwidegap
[1] 17.838066 18.601795 14.075963 21.714907 7.029929 9.004500 15.560914

$widestgap
[1] 21.71491

$sindex
[1] 10.22678

$corrected.rand
NULL

$vi
NULL

[1] "external clustering validation (corrected.rand, vi)"
[1] "Corrected Rand index"

[1] 0.149149
[1] "Meila variation of information (VI) index"
[1] 2.080936

4. Correlograms for MALDI_TOF spectra

4.1. Visual_CorrDistM

Visual_CorrDistM is based on functions for computing and visualizing distance matrix and correlograms

@param df_m: dataframe containing peaks and metadata
@param dist: distances ("euclidean", default value), "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski", "pearson", "spearman" or "kendall".
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie", "Genre","Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte"
@param value: value of catVar1 "Lactobacillus" ("Genre"), Taxonomie ("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition") ,...
@param Visual: correlogram, "Corr", (default value) or distance matrix figure ("Dist") (this second option is more general because it includes correlation, see argument dist)
@param CorrFig: correlogram type, "circle", "square", "ellipse", "number", "pie", "shade" and "color" (default value)
@param Ord: correlogram arrangement methods (ord="FPC", default value), "AOE", "hclust"
@param layout: correlogram layout: "full", "upper", "lower" (layout="upper", default value)
@param pv: boolean variable including or not probability values. (pv=TRUE, default value)
@param sig: significance level (sig="0.05", default value)
@return figures and statistics
@examples Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus"),
Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", VisualM="dist")
Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", VisualM="dist", dist="pearson")

source: https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html
https://www.rdocumentation.org/packages/corrplot/versions/0.84/topics/corrplot

Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus")

Output:
Correlation Matrix

	2018.79282168279	2035.60098965176	2051.55385901724	2067.95712274841	2083.58086146253
2018.79282168279	1.00	0.91	0.93	-0.29	0.10
2035.60098965176	0.91	1.00	0.95	-0.43	-0.06
2051.55385901724	0.93	0.95	1.00	-0.28	0.14
2067.95712274841	-0.29	-0.43	-0.28	1.00	0.88
2083.58086146253	0.10	-0.06	0.14	0.88	1.00
2106.58686543244	0.94	0.90	0.94	-0.14	0.27

	2106.58686543244	2124.50092568671	2147.40266739652	2166.00972951332	2182.83406086648
2018.79282168279	0.94	0.92	0.36	0.23	0.81
2035.60098965176	0.90	0.93	0.23	0.13	0.77
2051.55385901724	0.94	0.97	0.39	0.25	0.88
2067.95712274841	-0.14	-0.23	0.69	0.81	0.18
2083.58086146253	0.27	0.19	0.86	0.90	0.56
2106.58686543244	1.00	0.97	0.49	0.38	0.90

	2205.99202292547	2223.4110890903	2242.89870280795	2258.16638713302	2261.88048340716
2018.79282168279	0.91	0.78	-0.23	-0.14	0.15
2035.60098965176	0.88	0.69	-0.38	-0.22	0.14
2051.55385901724	0.96	0.84	-0.23	-0.10	0.22

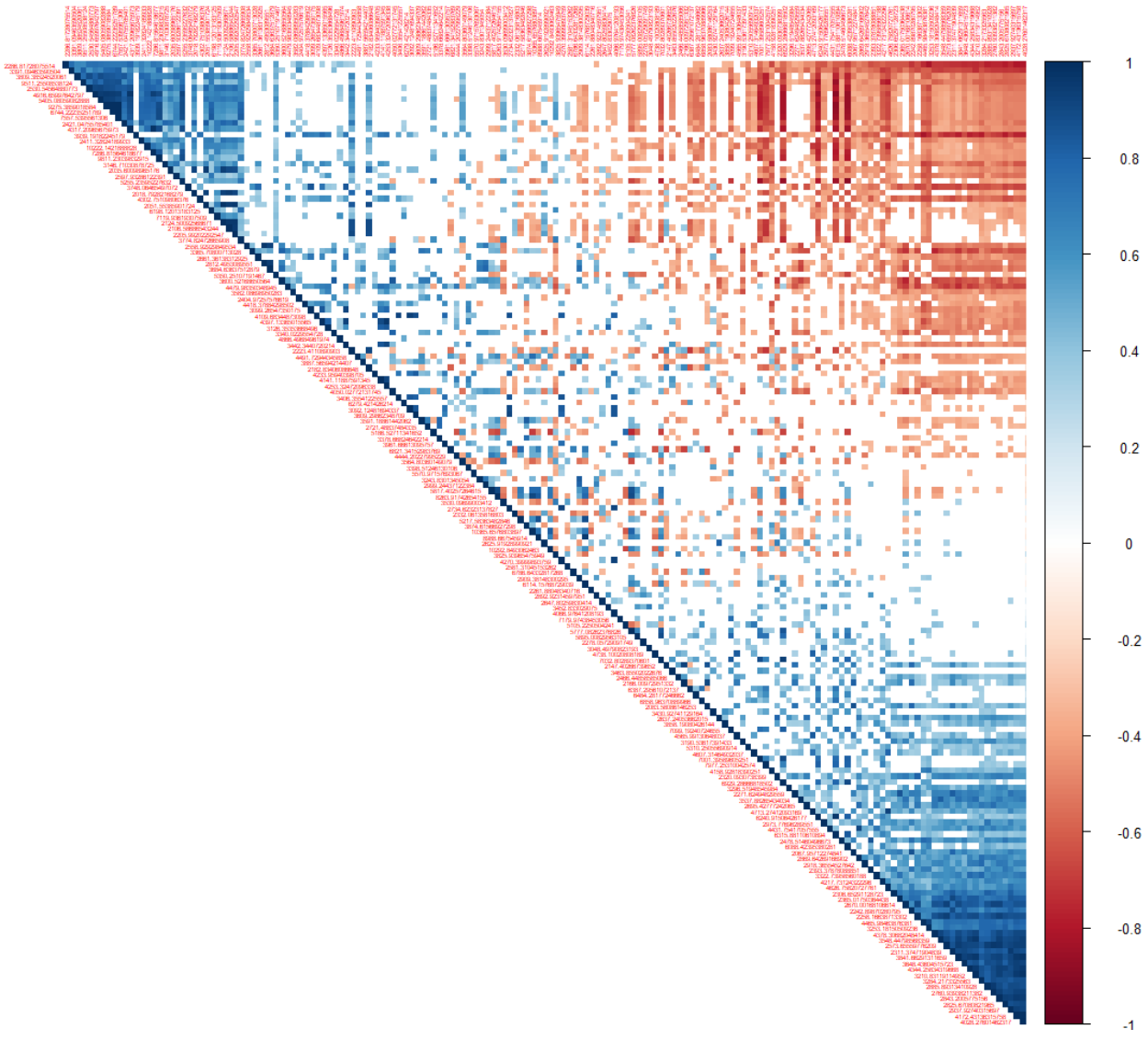
2067.95712274841	-0.19	0.07	0.87	0.49	0.18
2083.58086146253	0.23	0.47	0.79	0.45	0.30
2106.58686543244	0.96	0.89	-0.10	-0.06	0.21

2271.62494829559	2278.05729091749	2286.81728075514	2306.65291128723	2311.37471904839
2018.79282168279	0.13	-0.33	0.61	-0.20	-0.24
2035.60098965176	0.02	-0.36	0.64	-0.33	-0.38
2051.55385901724	0.15	-0.28	0.57	-0.22	-0.24
2067.95712274841	0.67	0.47	-0.55	0.66	0.79
2083.58086146253	0.73	0.37	-0.31	0.59	0.71
2106.58686543244	0.25	-0.24	0.52	-0.14	-0.12

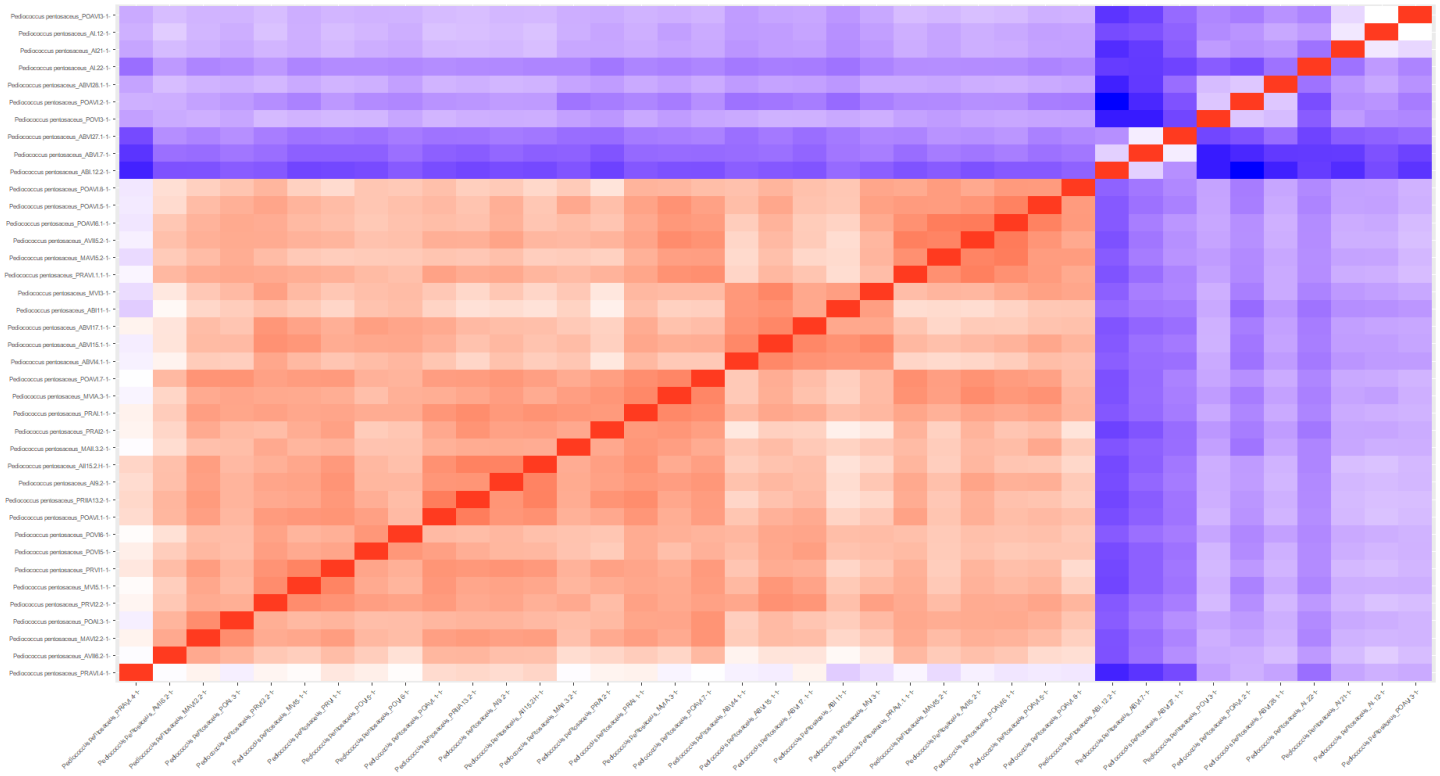
2320.0930738399	2332.06135816803	2365.01750364438	2393.37878088851	2404.97257576619
2018.79282168279	0.14	0.39	-0.10	-0.40	-0.10
2035.60098965176	0.09	0.36	-0.24	-0.43	-0.14
2051.55385901724	0.20	0.44	-0.13	-0.32	-0.13
2067.95712274841	0.34	0.07	0.70	0.65	-0.06
2083.58086146253	0.45	0.31	0.66	0.54	-0.09
2106.58686543244	0.21	0.42	0.00	-0.27	-0.12

2411.32824189933	2421.04755785401	2466.44858585066	2478.51460496673	2530.54564880773
2018.79282168279	0.70	0.78	0.10	-0.28	0.72
2035.60098965176	0.68	0.76	0.08	-0.35	0.74
2051.55385901724	0.69	0.83	0.10	-0.23	0.79
2067.95712274841	-0.42	-0.48	0.45	0.66	-0.40
2083.58086146253	-0.10	-0.09	0.48	0.56	-0.06
2106.58686543244	0.70	0.75	0.23	-0.16	0.72

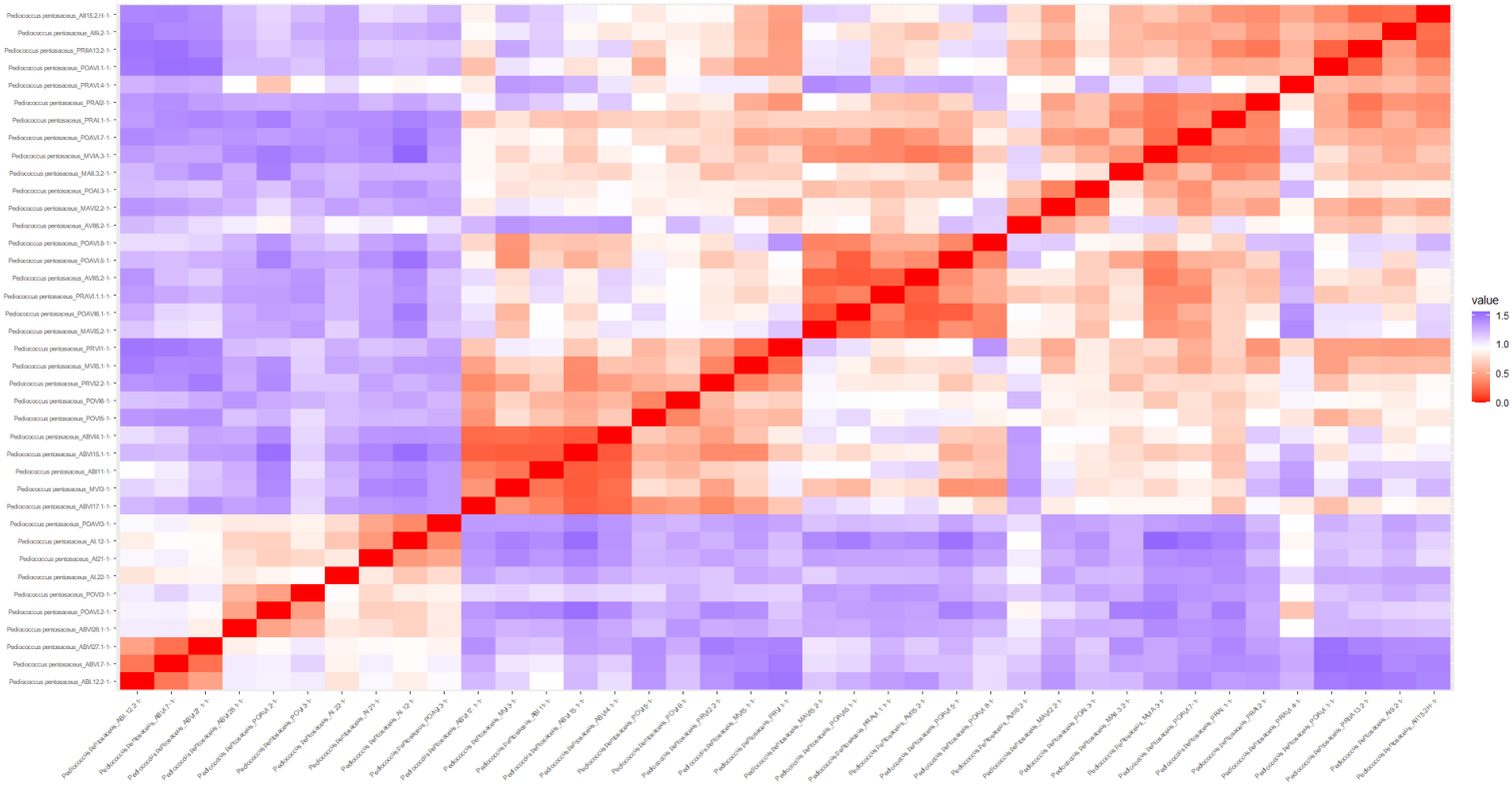
.....shortened output.....



Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", VisualM="dist")



Visual_CorrDistM (df_Peaks, varCat1="Taxonomie", value="Pediococcus pentosaceus", VisualM="dist", dist="pearson")



5. Principal Component Analysis and clustering of MALDI_TOF spectra

5.1. PCA_Clus

Principal Component Analysis and clustering of Maldi_Tof spectra

@param df_m: dataframe containing peaks and metadata
@param dist: distances, "euclidean", (default value), "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski", "pearson", "spearman" or "kendall".
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie", "Genre", "Date.d.analyse", "Origine","Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),…
@param meth: clustering method, ward(default value), "average" ,"single","complete'
@param graph: visual analysis, "dendf", "dendh", (dendograms) "factorMapf", "factorMaph", "factorMapClus", (factor maps) (default value, graph="factorMapClus")
@param pc: number of principal components (pc=3, default value)
@return figures and statistics
@examples pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus"),
pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="dendh")
pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="dendf"),
pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="factorMapf")
source: https://www.rdocumentation.org/packages/FactoMineR/versions/2.2/topics/PCA
https://www.rdocumentation.org/packages/FactoMineR/versions/2.2/topics/HCPC
http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/117-hcpc-hierarchical-clustering-on-principal-components-essentials/
https://rpkgs.datanovia.com/factoextra/
http://factominer.free.fr/factomethods/hierarchical-clustering-on-principal-components.html

pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus")

Output:

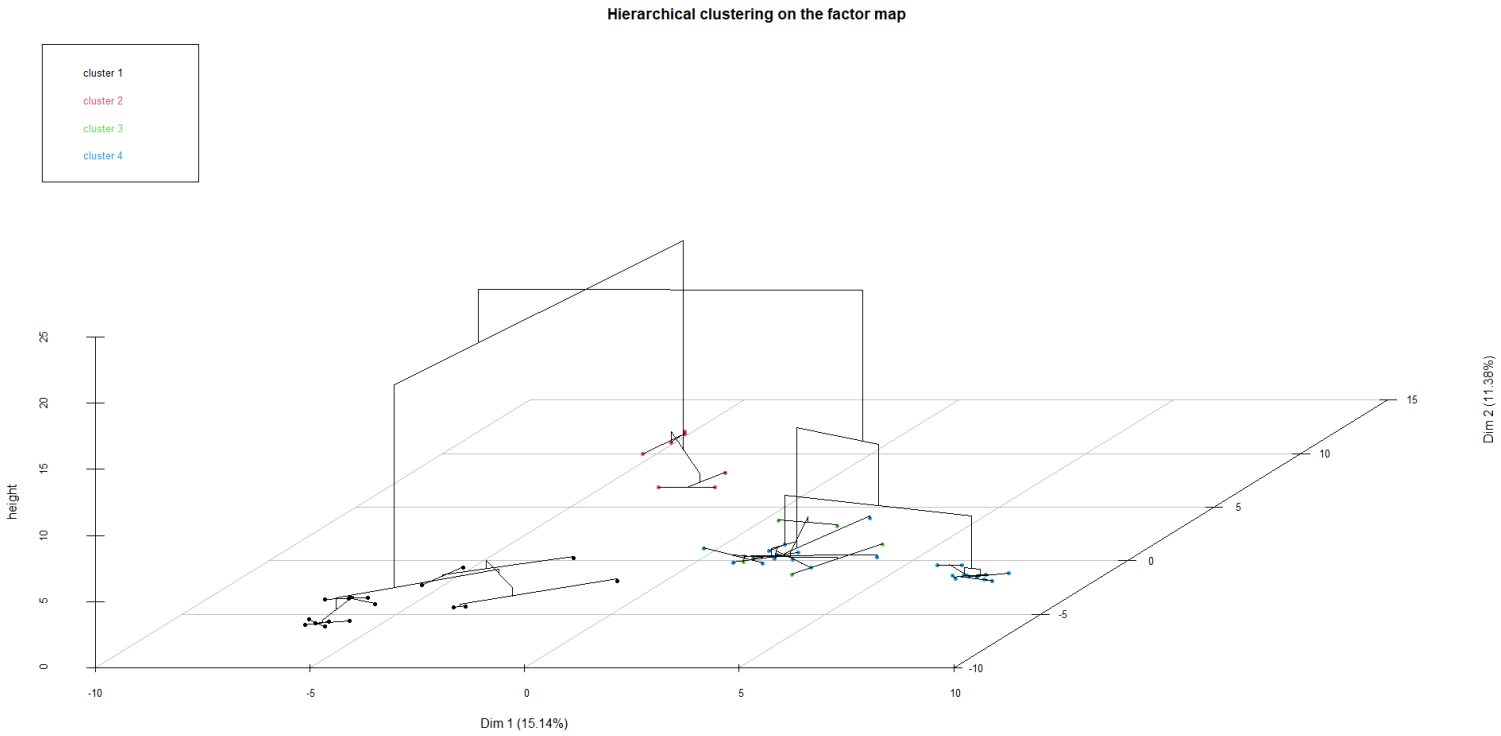
[1] "contribution of variables for each cluster"
$`1`

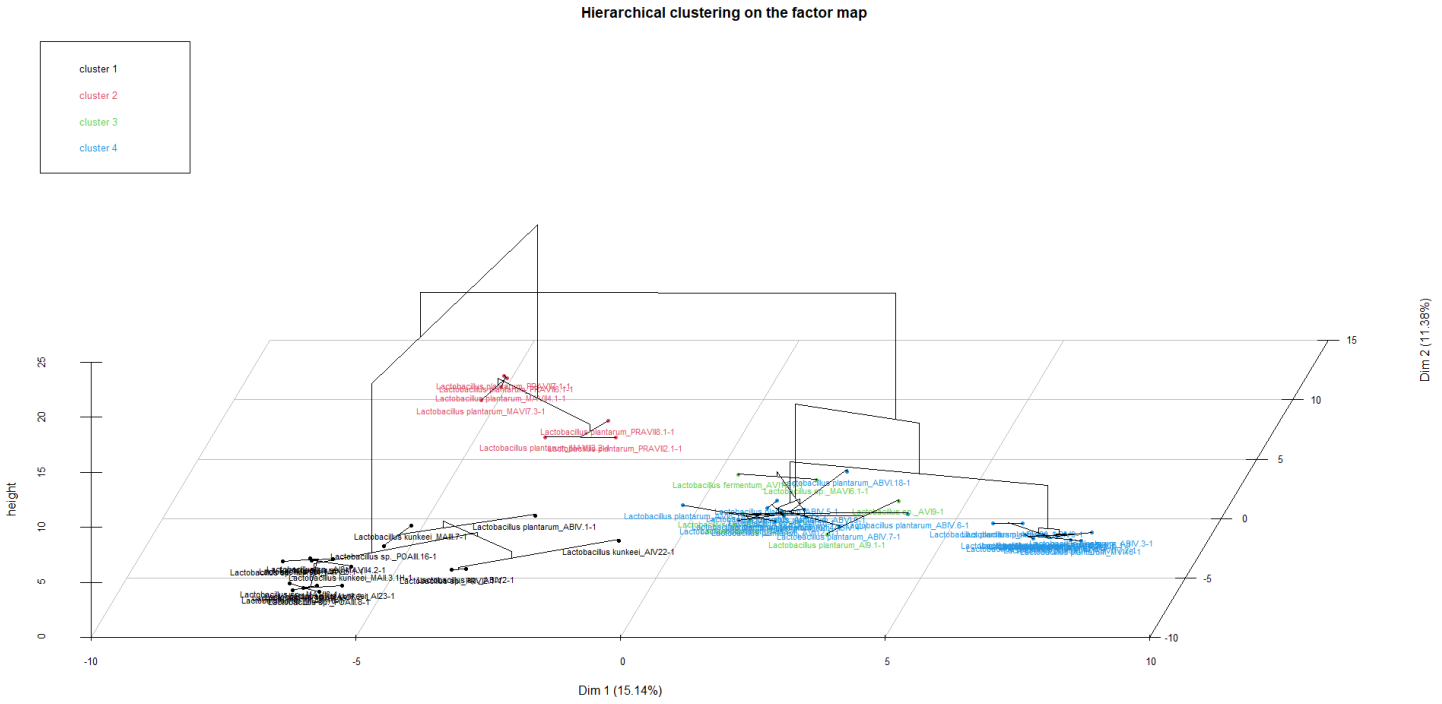
	v.test	Mean in category	Overall mean	sd in category	Overall sd	p.value
9275.3859018584	7.016901	9.113426e-04	2.872487e-04	2.543822e-04	4.354899e-04	2.268425e-12
10222.1421888828	6.260484	2.745480e-04	1.219477e-04	1.012603e-04	1.193496e-04	3.837844e-10
10292.8493062463	6.104863	2.435175e-04	8.805615e-05	1.275495e-04	1.246867e-04	1.028890e-09
6821.34152983769	5.954612	8.517342e-04	3.371016e-04	4.425406e-04	4.231724e-04	2.606898e-09
5186.52711341652	5.933142	2.650866e-04	1.150800e-04	9.924594e-05	1.237938e-04	2.971914e-09
4344.25834319668	5.675180	2.798768e-04	1.371277e-04	8.857747e-05	1.231593e-04	1.385432e-08
5105.2250504241	5.368627	1.789227e-04	1.043075e-04	5.816985e-05	6.805145e-05	7.933832e-08
9811.23039832915	5.217313	8.499936e-05	3.680865e-05	3.206304e-05	4.522619e-05	1.815378e-07
7557.5395561306	5.143829	1.413613e-04	6.339382e-05	7.749443e-05	7.421651e-05	2.691953e-07
6744.22235251789	4.876365	4.332199e-04	1.987914e-04	3.016745e-04	2.353899e-04	1.080588e-06
4378.30682048414	4.799099	7.080005e-04	3.220108e-04	4.981437e-04	3.938125e-04	1.593810e-06
7032.80289370601	4.699584	2.360603e-04	1.262617e-04	1.157892e-04	1.143960e-04	2.606918e-06

```
3406.35541225557  4.155175    4.881689e-04 2.500485e-04   2.989195e-04 2.805956e-04 3.250384e-05
10365.6576803897  4.072204    6.169871e-05 3.432341e-05   2.372842e-05 3.291568e-05 4.657032e-05
5217.58363482846  3.941380    1.531350e-04 8.342728e-05   7.286626e-05 8.659770e-05 8.101421e-05
3365.70800713028  3.862362    3.405563e-04 1.539493e-04   3.304499e-04 2.365639e-04 1.122959e-04
7286.81564618677  3.693756    2.772442e-04 1.131997e-04   3.102364e-04 2.174538e-04 2.209656e-04
3378.66824642214  3.656797    3.909689e-04 1.892685e-04   3.217762e-04 2.700718e-04 2.553862e-04
6786.64332817268  3.521235    4.826700e-04 2.768113e-04   4.166497e-04 2.862514e-04 4.295418e-04
4317.20965675973  3.496074    6.028930e-04 2.980956e-04   6.214058e-04 4.268783e-04 4.721581e-04
2320.0930738399   3.373547    2.661921e-04 1.357076e-04   2.898871e-04 1.893850e-04 7.420646e-04
6858.96370889966  3.269224    3.965194e-04 2.515290e-04   3.332777e-04 2.171540e-04 1.078427e-03
2083.58086146253  3.191997    8.027818e-04 5.510887e-04   4.884487e-04 3.860844e-04 1.412928e-03
6279.421426214    2.941411    3.361422e-04 2.372500e-04   1.811478e-04 1.646190e-04 3.267210e-03
4397.13365015565  2.638642    2.332052e-04 1.524672e-04   8.577571e-05 1.498204e-04 8.323872e-03
2393.37878088851  2.596518    1.473276e-04 8.977535e-05   1.645571e-04 1.085287e-04 9.417384e-03
```

.....shortened output.....

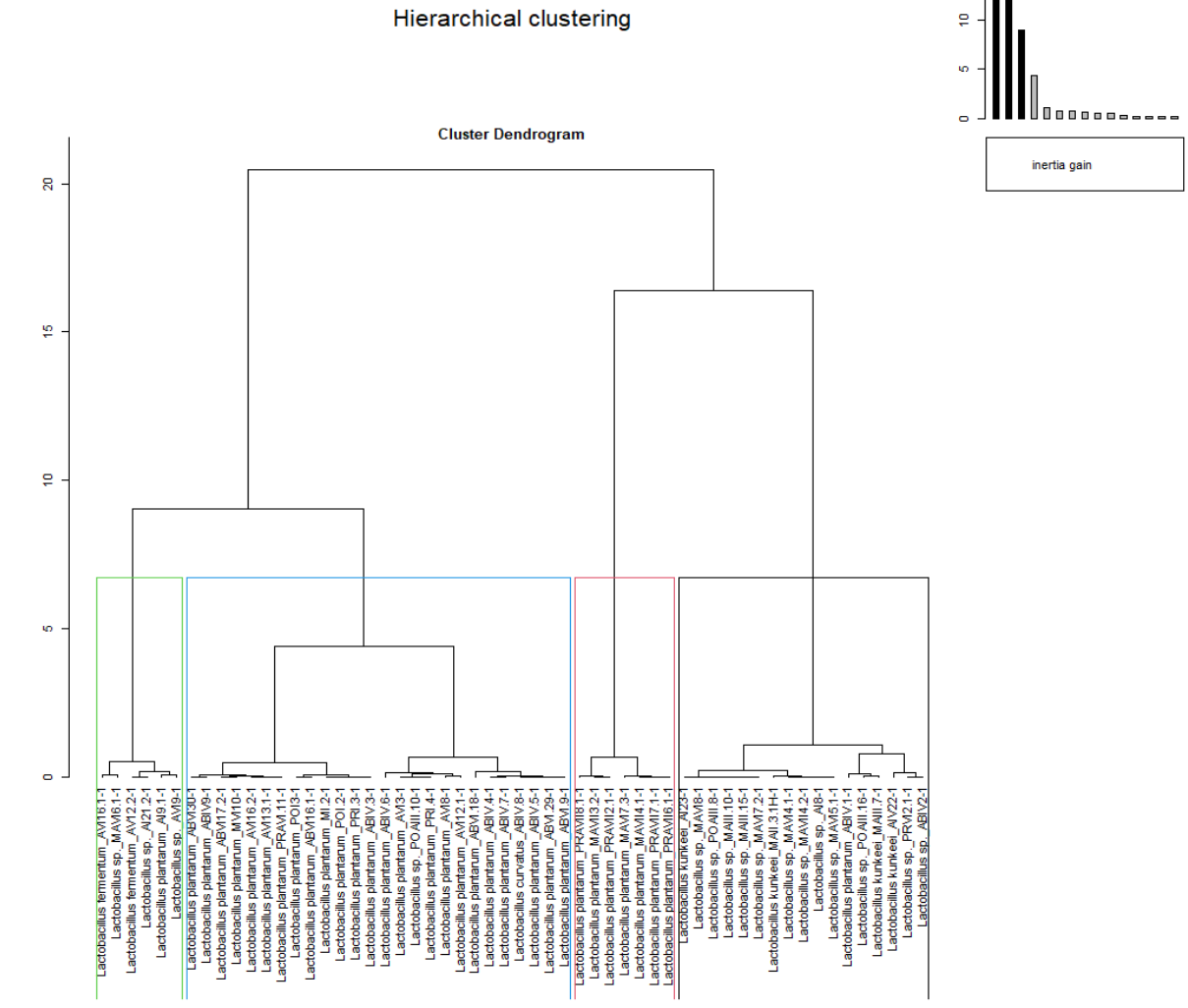
```
[1] "contribution  of principal components  for each cluster"
$`1`
      v.test Mean in category  Overall mean sd in category Overall sd     p.value
Dim.2 -4.365895       -3.863117 -7.432051e-16       1.831196   4.332495 1.266031e-05
Dim.1 -5.412847       -5.525758  5.144694e-16       1.603985   4.998499 6.203042e-08

$`2`
      v.test Mean in category  Overall mean sd in category Overall sd     p.value
Dim.2  6.163973        9.527227 -7.432051e-16       2.074991   4.332495 7.094185e-10
Dim.1 -2.356219       -4.201679  5.144694e-16       1.177872   4.998499 1.846204e-02

$`3`
      v.test Mean in category Overall mean sd in category Overall sd     p.value
Dim.3 6.194998        8.625274 1.298812e-15       1.903931   3.576874 5.828569e-10

$`4`
      v.test Mean in category Overall mean sd in category Overall sd     p.value
Dim.1  5.982709        4.331418 5.144694e-16       2.690896   4.998499 2.194566e-09
Dim.3 -2.171633       -1.125078 1.298812e-15       2.226520   3.576874 2.988339e-02
```



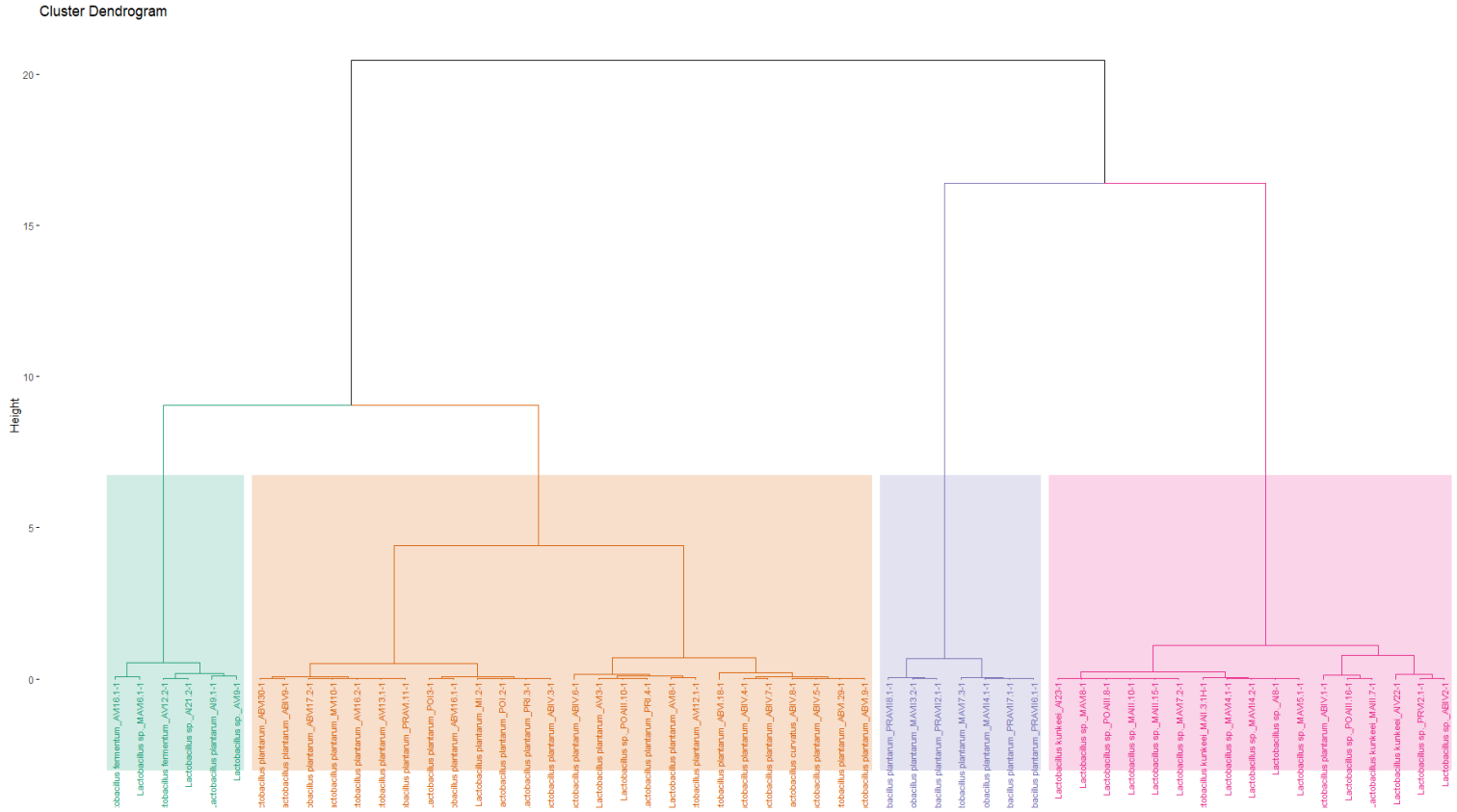


pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="dendh")

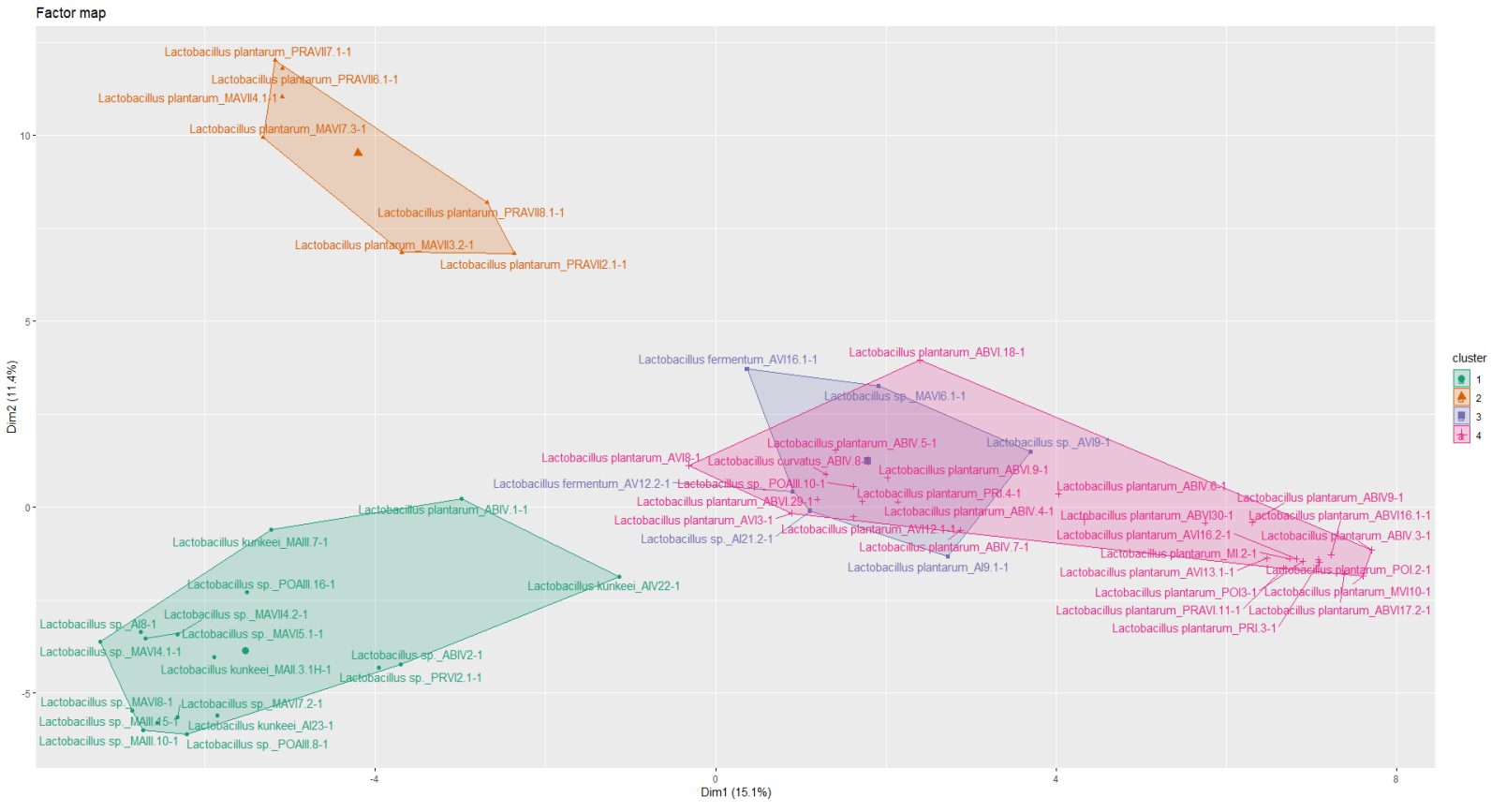
(excluded non-visual output)



pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="dendf")



pc<-PCA_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus", graph="factorMapf")



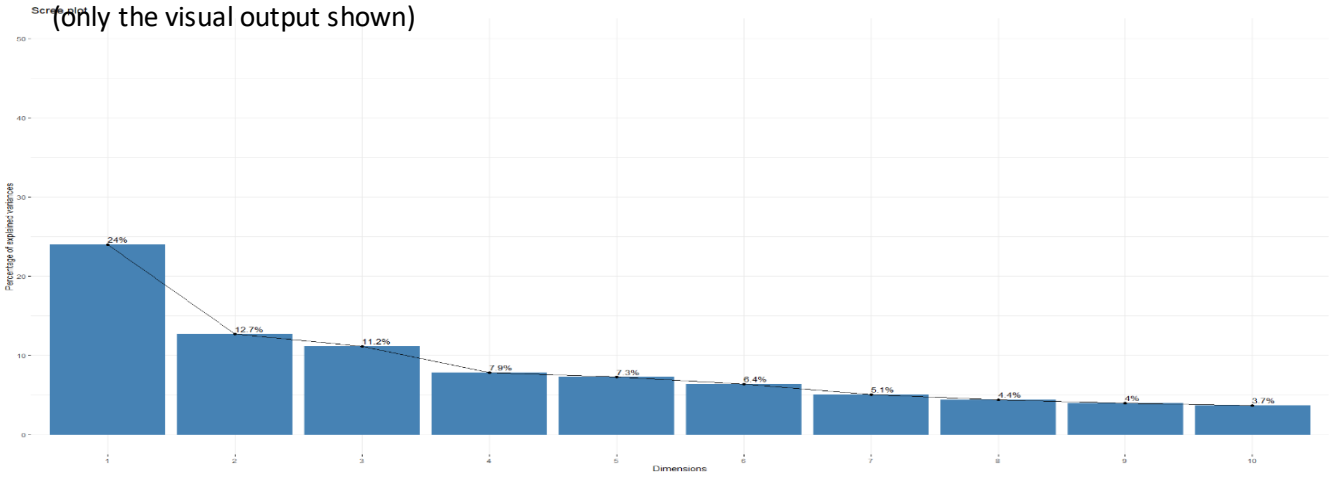
5.2. SPCA

Principal Component Analysis of Maldi_Tof spectra with external clusters based on metadata information

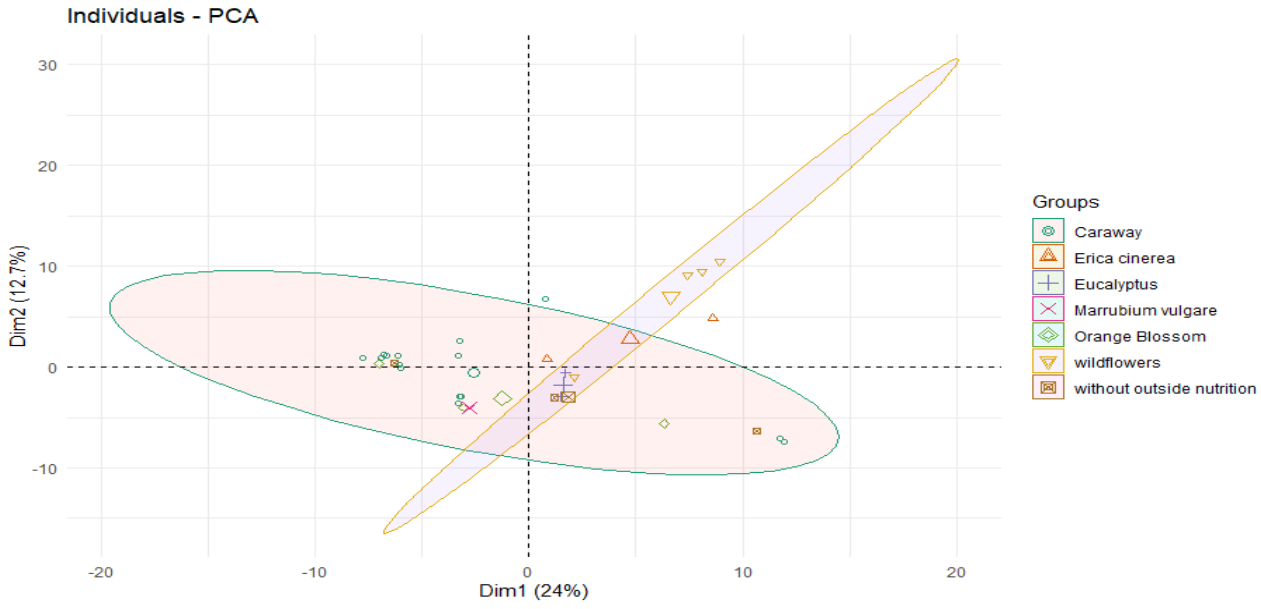
@param df_m: dataframe containing peaks and metadata
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie", "Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of catVar1 "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param varCat2: categorical variable for partitioning the chosen isolates, Taxonomie, Genre, Date.d.analyse, Origine, Ruche, Nutrition, Date.de.récolte, Lieu.de.la.ruche
@param contDim: graph and statistics for PC contributions (default, contDim =TRUE)
@param contVar: graph and statistics for variable contributions(default, contVar=FALSE)
@param contInd: graph and statistics for isolate contributions (default, contInd= TRUE)
@examples hg<-SPCA(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")

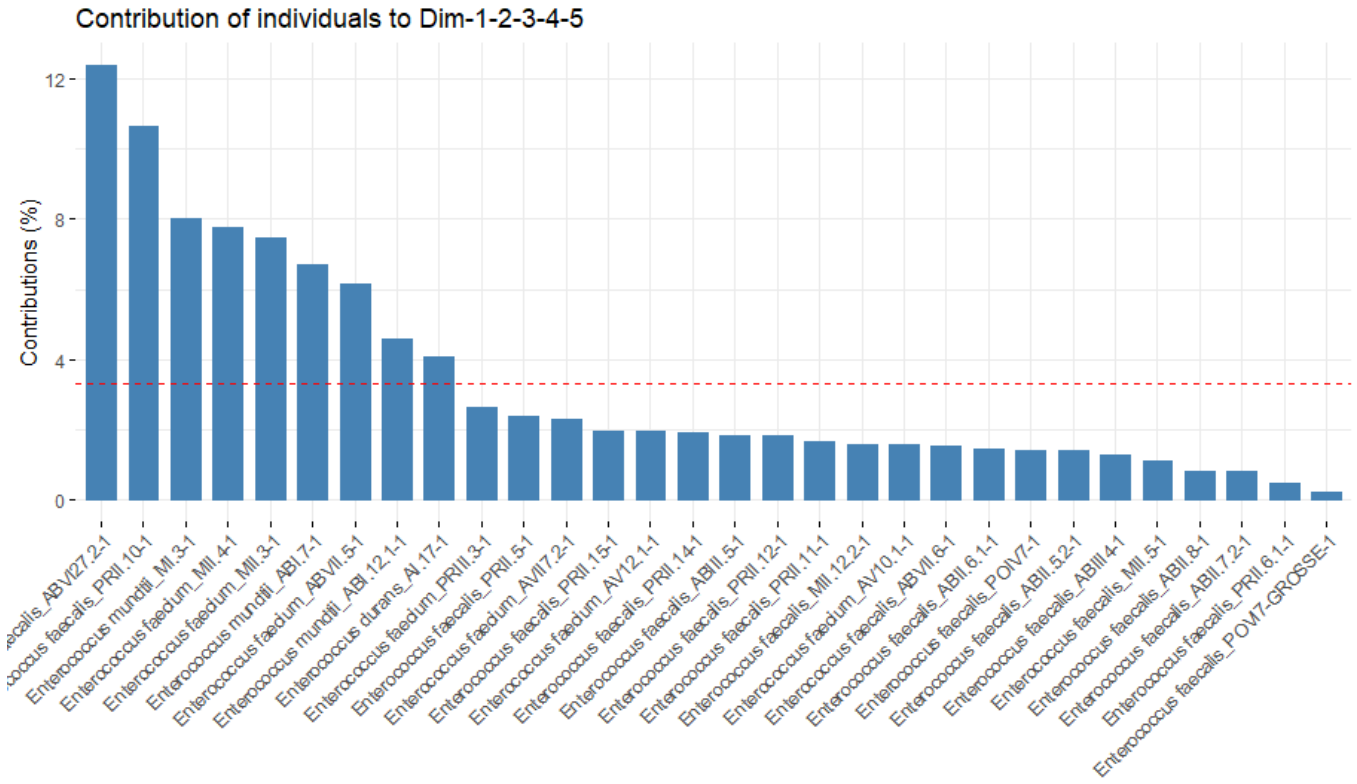
SPCA<- function(df_m, varCat1, value, varCat2, contDim=TRUE, contVar= FALSE, contInd=FALSE)

(only the visual output shown)



hg<-SPCA(df_Peaks, varCat1="Genre",value="Enterococcus", varCat2="Nutrition",contDim=FALSE, contVar= FALSE,contInd=TRUE) (only the visual output shown)





6. Multidimensional scaling and clustering of MALDI_TOF spectra

6.1. MDS_Clus

MDS_Clus, function for Multidimensional scaling and kmeans-based analysis of Maldi_Tof spectra

@param df_m: dataframe containing peaks and metadata
@param dist: distances: "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie","Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte", "Lieu.de.la.ruche"
@param value: level of the chosen categorical variable as catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition"),...
@param grah: graphs: lab_mdsclus,(default value) lb_mds, mdsclaing
@return figures and statistics
@examples g<-MDS_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus")

source: https://rstudio-pubs-static.s3.amazonaws.com/274936_050c742fb3514bbaa87ce6ee2686af8c.html
http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/122-multidimensional-scaling-essentials-algorithms-and-r-code/
http://ugrad.stat.ubc.ca/R/library/mva/html/cmdscale.html

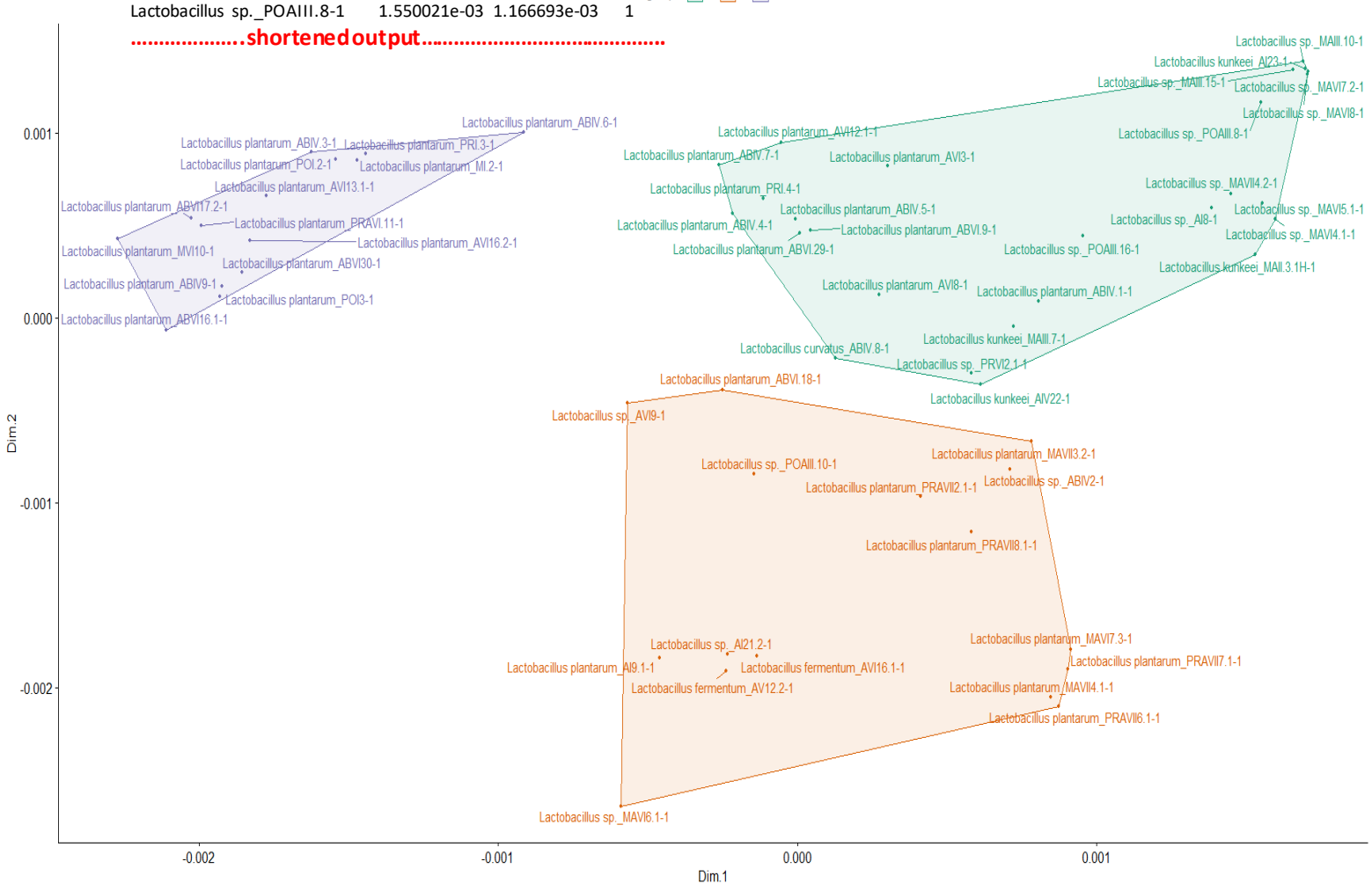
g<-MDS_Clus(df_Peaks, varCat1="Genre", value="Lactobacillus")

output:

	Dim.1	Dim.2	groups
Lactobacillus plantarum_ABIV.1-1	8.073724e-04	9.230089e-05	1
Lactobacillus plantarum_ABIV.3-1	-1.626189e-03	8.995464e-04	3
Lactobacillus plantarum_ABIV.4-1	-2.157681e-04	5.663535e-04	1

Lactobacillus plantarum_ABIV.5-1 -7.135335e-06 5.353676e-04 1
Lactobacillus plantarum_ABIV.6-1 -9.152847e-04 1.004312e-03 3
Lactobacillus plantarum_ABIV.7-1 -2.614522e-04 8.268632e-04 1
Lactobacillus curvatus_ABIV.8-1 1.267334e-04 -2.178710e-04 1
Lactobacillus plantarum_ABVI.18-1 -2.491353e-04 -3.890558e-04 2
Lactobacillus plantarum_ABVI.29-1 7.677427e-06 4.594310e-04 1
Lactobacillus plantarum_ABVI.9-1 4.335097e-05 4.721244e-04 1
Lactobacillus kunkeei_MAII.3.1H-1 1.531006e-03 3.411630e-04 1
Lactobacillus sp._MAIII.10-1 1.691986e-03 1.385961e-03 1
Lactobacillus sp._MAIII.15-1 1.657554e-03 1.341994e-03 1
Lactobacillus kunkeei_MAIII.7-1 7.231229e-04 -4.565799e-05 1
Lactobacillus plantarum_MI.2-1 -1.471482e-03 8.523043e-04 3
Lactobacillus sp._POAIII.10-1 -1.439868e-04 -8.438319e-04 2
Lactobacillus sp._POAIII.16-1 9.544077e-04 4.409873e-04 1
Lactobacillus sp._POAIII.8-1 1.550021e-03 1.166693e-03 1

.....shortened output.....



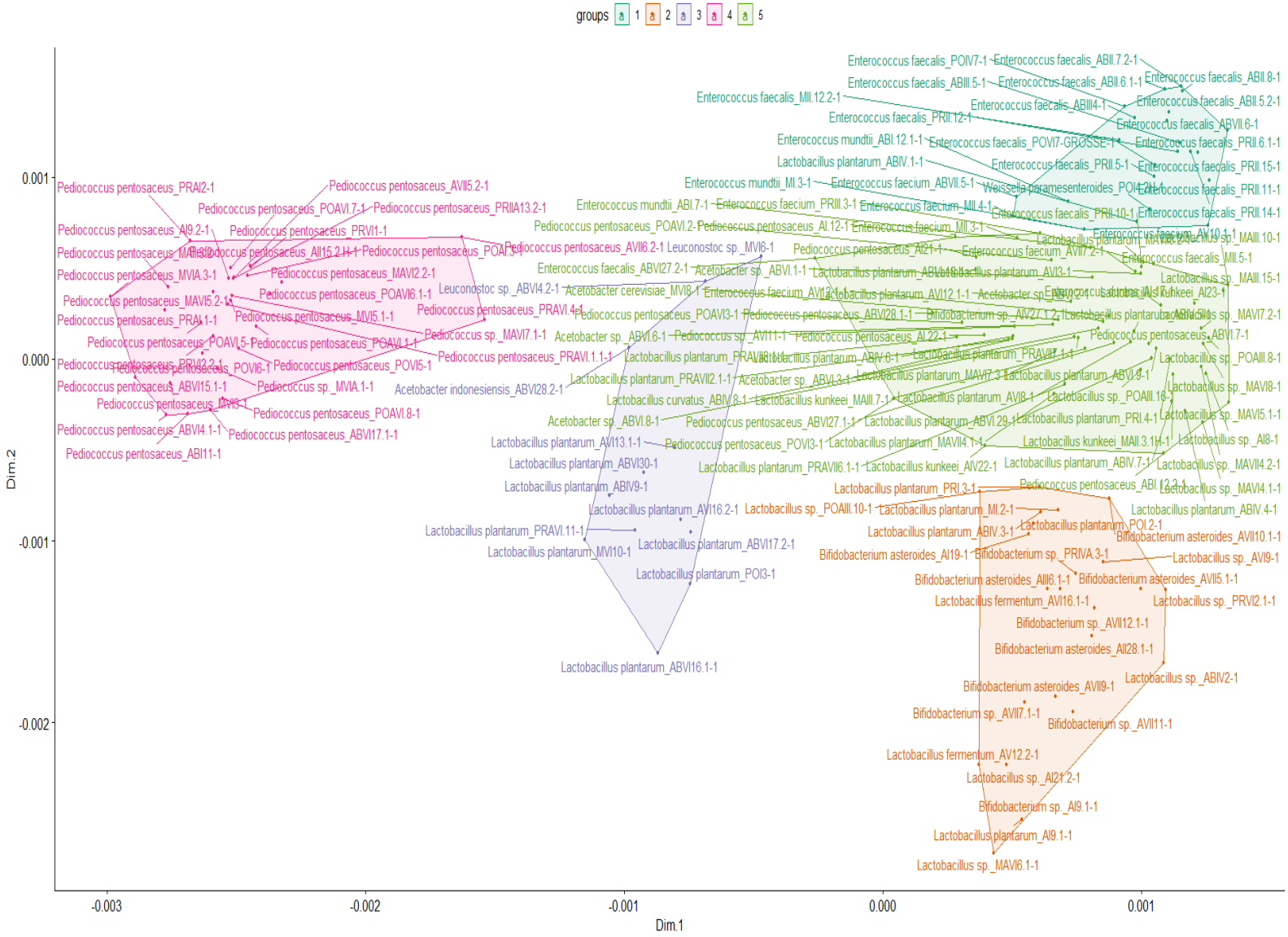
g<-MDS_Clus(df_Peaks, varCat1="Taxonomie", value="All", nc=5)

output:

Dim.1 Dim.2 groups
Pediococcus pentosaceus_AI.12-1 2.783290e-04 6.720991e-04 5
Enterococcus durans_AI.17-1 1.074552e-03 2.928558e-04 5
Pediococcus pentosaceus_AI.22-1 3.911749e-04 1.297149e-04 5
Enterococcus mundtii_ABI.12.1-1 7.160698e-04 8.675265e-04 1
Pediococcus pentosaceus_ABI.12.2-1 1.085123e-03 -5.198231e-04 5
Enterococcus mundtii_ABI.7-1 8.018773e-04 6.104123e-04 5

Enterococcus faecalis_ABII.5.2-1	1.105291e-03	1.356064e-03	1
Enterococcus faecalis_ABII.6.1-1	1.090702e-03	1.486138e-03	1
Enterococcus faecalis_ABII.7.2-1	1.152658e-03	1.498682e-03	1
Enterococcus faecalis_ABII.8-1	1.158135e-03	1.474918e-03	1
Enterococcus faecalis_ABIII.5-1	1.146276e-03	1.190664e-03	1
Lactobacillus plantarum_ABIV.1-1	9.803381e-04	7.566789e-04	1
Lactobacillus plantarum_ABIV.3-1	6.084417e-04	-8.424467e-04	2
Lactobacillus plantarum_ABIV.4-1	1.166997e-03	-2.844821e-04	5
Lactobacillus plantarum_ABIV.5-1	9.696000e-04	1.512996e-04	5
Lactobacillus plantarum_ABIV.6-1	5.021067e-04	1.256638e-04	5

……shortened output……



6.2. SMDS

Multidimensional scaling and external cluster-based analysis of MALDI_TOF spectra

@param df_m: dataframe containing peaks and metadata
@param dist: distances: "euclidean" (default value), "maximum", "manhattan", "canberra", "binary" or "minkowski"
@param varCat1: categorical variable for choosing isolates, examples: "Taxonomie",

"Genre", "Date.d.analyse", "Origine", "Ruche", "Nutrition", "Date.de.récolte",
"Lieu.de.la.ruche"

@param value: level of the chosen categorical variable catVar1, examples: "Lactobacillus" ("Genre"), Taxonomie("Pediococcus pentosaceus"), "Erica cinerea" ("Nutrition") ,...

@param grah: graphs: "lab_mdsGroups" (default value), "mdsGroups"

@return figures and statistics

@examples

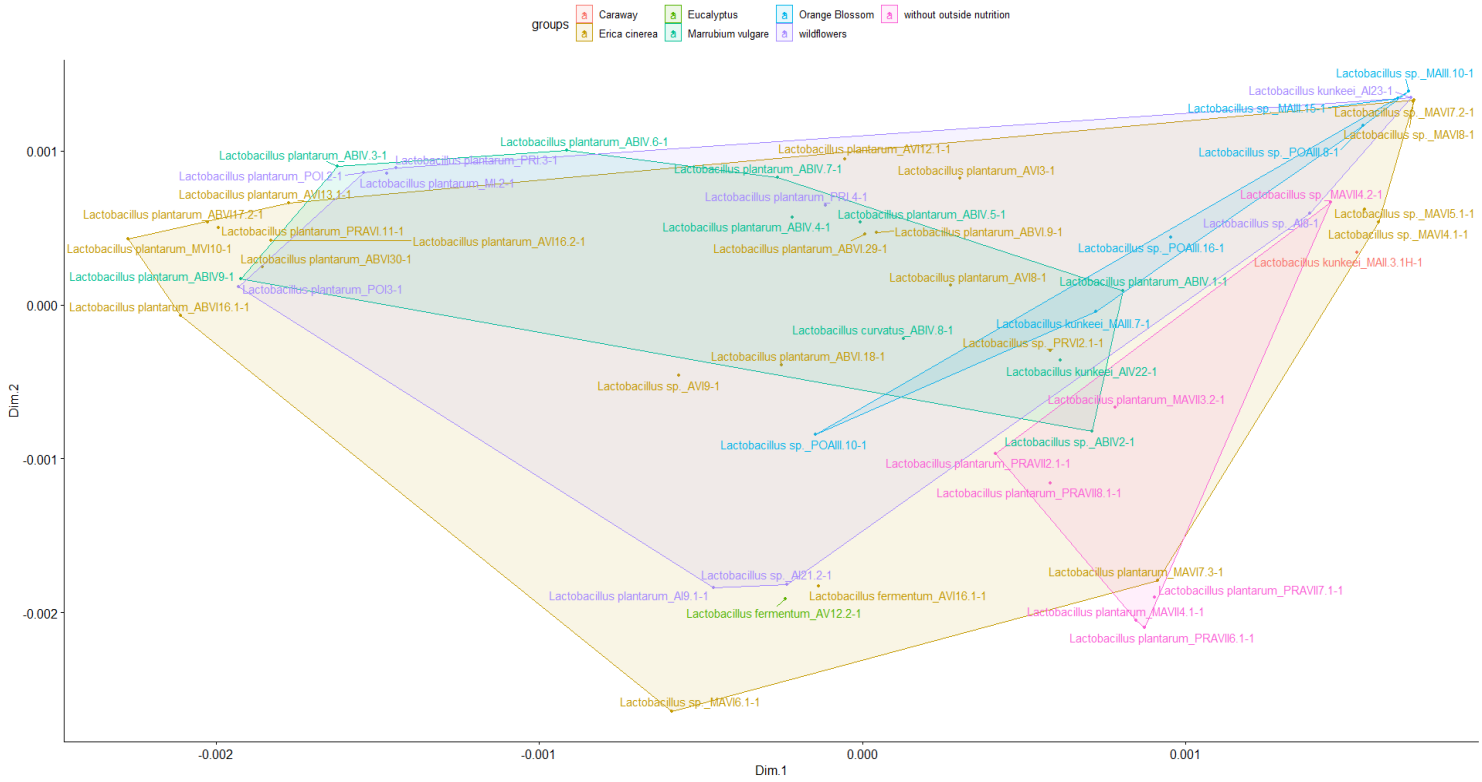
SMDS(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Ruche",dist="euclidean", grah="lab_mdsGroups")

SMDS(df_Peaks, varCat1="Genre", value="Lactobacillus" varCat2="Ruche",)

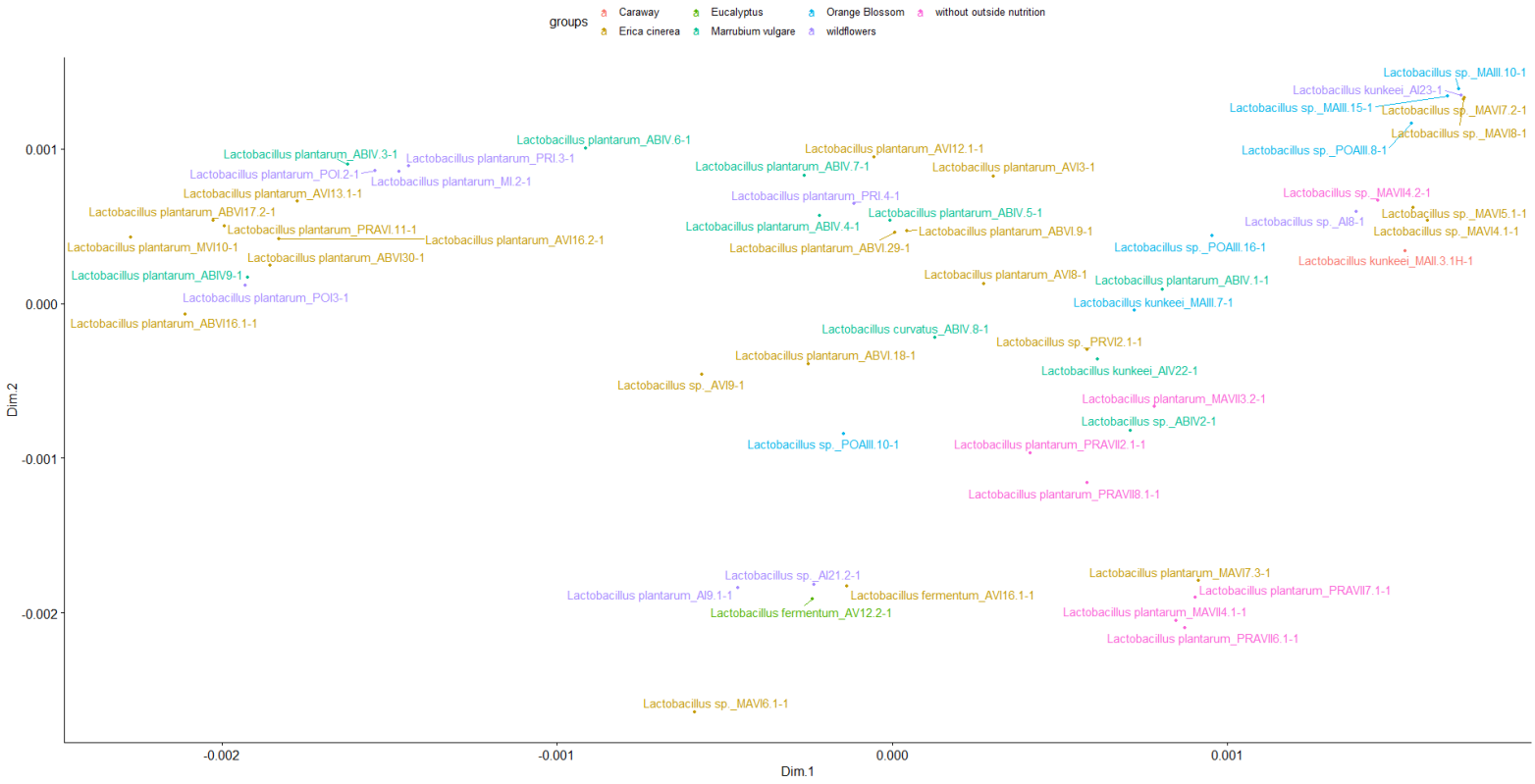
SMDS(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")

SMDS(df_Peaks, varCat1="Genre", value="All",varCat2="Taxonomie")

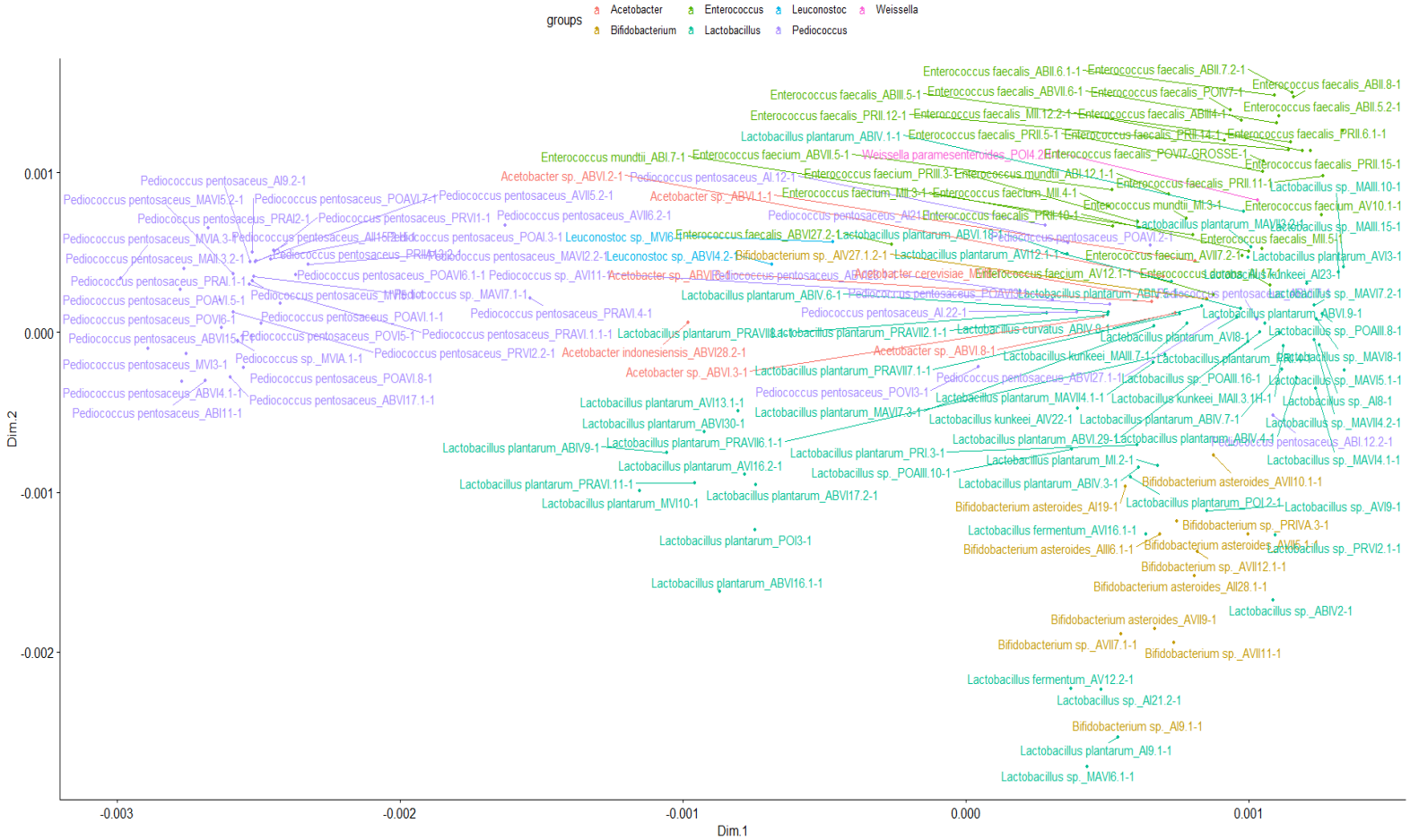
SMDS(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition")



SMDS(df_Peaks, varCat1="Genre", value="Lactobacillus", varCat2="Nutrition",grah="mdsGroups")



SMDS(df_Peaks, varCat1="Taxonomie", value="All", varCat2="Genre",grah="mdsGroups")



SMDS(df_Peaks, varCat1="Genre", value="All", varCat2="Taxonomie",grah="mdsGroups")