Exploratory analysis of EBI-Metagenomic portal contingency tables enriched with ENA metadata

Blaise T.F. Alako

Bari, June, 2017

Dataset: American Gut Project
The American Gut project is the largest crowdsourced citizen science project to date. Fecal, oral, skin, and other body site samples collected from thousands of participants represent the largest human microbiome cohort in existence. Detailed health and lifestyle and diet data associated with each sample is enabling us to deeply examine associations between the human microbiome and factors such as diet (from vegan to near carnivore and everything in between), season, amount of sleep, and disease states such as IBD, diabetes, or autism spectrum disorder-as well as many other factors not listed here. The American Gut project also encompasses the British Gut and Australian Gut projects, widening the cohort beyond North America. As the project continues to grow, we will be able to identify significant associations that would not be possible with smaller, geographically and health/disease status-limited cohorts. *ENA website (ERP012803)*

We will explore the data generated by the study above. The data is available at:
https://www.ebi.ac.uk/metagenomics/projects/ERP012803

Loading of R packages

```r
suppressWarnings(suppressMessages(require(stringr)))      # String format
suppressWarnings(suppressMessages(require(ade4)))         # Multivariate analysis
suppressWarnings(suppressMessages(require(ggplot2)))      # Fancy plotting
suppressWarnings(suppressMessages(require(grid)))         # Has the viewport function
suppressWarnings(suppressMessages(require(dplyr)))        # Data manipulation
suppressWarnings(suppressMessages(require(tidyr)))        # Data manipulation
suppressWarnings(suppressMessages(require(factoextra)))   # Visualize result of
suppressWarnings(suppressMessages(require(FactoMineR)))   # multivariate analysis
suppressWarnings(suppressMessages(require(XML)))
suppressWarnings(suppressMessages(require(xml2)))
suppressWarnings(suppressMessages(devtools::install_github("hrbrmstr/xmlview")))
suppressWarnings(suppressMessages(require(xmlview)))
suppressWarnings(suppressMessages(require(xml2)))
suppressWarnings(suppressMessages(require(purrr)))
suppressWarnings(suppressMessages(require(surveillance)))
suppressWarnings(suppressMessages(require('gplots')))
suppressWarnings(suppressMessages(require(webshot)))
```

Obtain input data for analysis

```r
# Get the analysis summary file
ERP012803 <- "https://www.ebi.ac.uk/metagenomics/projects/ERP012803/download/2.0/export?contentType=tex
ERP012803.meta <- "https://www.ebi.ac.uk/metagenomics/projects/ERP012803/overview/doExport";
gutproject <- tbl_df(read.delim(file=ERP012803))
meta <-  tbl_df(read.csv(file=ERP012803.meta))
meta.MG <- meta %>%   mutate(Sample.Description=gsub(" ","_",Sample.Description),
                         Sample.Description=gsub("American_Gut_Project_","", Sample.Description),
                         Sample.Description=gsub("American_","", Sample.Description)
                         ) %>% select(Sample.Description, Sample.ID, Run.ID)
```

```
#############################################################
#Extracts extensive information about the sample from the ENA
#############################################################
feature_selection <- c('sample_id','alcohol_frequency','cosmetics_frequency','diet_type',
                       'high_fat_red_meat_frequency','level_of_education','probiotic_frequency',
                       'red_meat_frequency','sex','teethbrushing_frequency','liver_disease',
                       'lung_disease','thyroid','alcohol_types_red_wine','alcohol_types_white_wine',
                       'cardiovascular_disease','fruit_frequency','meat_eggs_frequency',
                       'multivitamin','pregnant','alcohol_consumption','alcohol_types_sour_beers',
                       'diabetes','bmi_cat','cat','kidney_disease','nail_biter','tonsils_removed')

sub_feature <- c('sample_id','alcohol_frequency', 'diet_type','sex','fruit_frequency',
                 'meat_eggs_frequency','multivitamin','alcohol_consumption','bmi_cat','diabetes',
                 'liver_disease','lung_disease','thyroid','cardiovascular_disease',
                 'pregnant','kidney_disease'
                 )

extractSampleInfo <- function(x){
  tag <- xml_text(xml_find_all(x, xpath='.//TAG'))
  tag <- as.character(tag)
  value <- xml_text(xml_find_all(x, xpath='.//VALUE'))
  value <- as.character(value)
  id <- xml_text(xml_find_all(x, xpath='.//PRIMARY_ID'))
  id <- as.character(id)
  sampleInfo <- tbl_df(data.frame(id=id, tag=tag,value=value))
  #Select a subset of features
  sampleInfo <- sampleInfo %>% filter(tag %in% feature_selection) #%>% filter(!value=='Unknown')
  return(sampleInfo)
}

constructQuery <- function( accs=accs){
  info <- paste("Processing ... ", length(accs), ' BiosampleID' , sep="")
  print(info)
  accs <- paste(accs,collapse = ",")
  url <- paste("https://www.ebi.ac.uk/ena/data/view/", accs, "&display=xml", sep="")
  return(url)
}

getXML <- function(url=url){
  xml <- read_xml(url)
  return(xml)
}

retrieveXmlContent <- function(accs=accs){
  query <- constructQuery(accs=accs)
  xmlContent <- getXML(url=query)
  return(xmlContent)
}
# What information are capture for each biosample

biosample_id <- as.character(meta$Sample.ID[1])
biosample.xml <- retrieveXmlContent(accs=biosample_id)
```

```
## [1] "Processing ... 1 BiosampleID"
```

Uncomment the following code to view a sample ENA SAMPLE XML file

```
#xml_view(biosample.xml, add_filter = TRUE)
#webshot(xml_view(biosample.xml, add_filter = TRUE), "r.png")
# NB: look for TAG that contains alcohol: (.//TAG[contains(.,'alcohol')]) in the XPath box
# How many feature are capture for this sample?
xml_find_all(biosample.xml, './/TAG', ns=xml2::xml_ns(biosample.xml))
```

```
## {xml_nodeset (432)}
##  [1] <TAG>acne_medication</TAG>
##  [2] <TAG>acne_medication_otc</TAG>
##  [3] <TAG>add_adhd</TAG>
##  [4] <TAG>age_cat</TAG>
##  [5] <TAG>age_corrected</TAG>
##  [6] <TAG>age_years</TAG>
##  [7] <TAG>alcohol_consumption</TAG>
##  [8] <TAG>alcohol_frequency</TAG>
##  [9] <TAG>alcohol_types_beercider</TAG>
## [10] <TAG>alcohol_types_red_wine</TAG>
## [11] <TAG>alcohol_types_sour_beers</TAG>
## [12] <TAG>alcohol_types_spiritshard_alcohol</TAG>
## [13] <TAG>alcohol_types_unspecified</TAG>
## [14] <TAG>alcohol_types_white_wine</TAG>
## [15] <TAG>allergic_to_i_have_no_food_allergies_that_i_know_of</TAG>
## [16] <TAG>allergic_to_other</TAG>
## [17] <TAG>allergic_to_peanuts</TAG>
## [18] <TAG>allergic_to_shellfish</TAG>
## [19] <TAG>allergic_to_tree_nuts</TAG>
## [20] <TAG>allergic_to_unspecified</TAG>
## ...
```

```
#There are 432 feature captured, we should select a subset for downstream analysis
```

Uncomment the following to retrieve metadata from ENA

```
#-------------------------------------------------------------
#------ The following is time consuming hence commented.
#------ The retrieved data is provided as an R object below
#-------------------------------------------------------------
# Since we have > 8000 sample to fetch extensive information,
# lets us process this in batch of 500

# accs <- split(meta$Sample.ID, ceiling(seq_along(meta$Sample.ID)/500))
# system.time(results<- lapply(accs, retrieveXmlContent))
#
# # Loop through the chunck in order to extract Sample Information
# # #################################################################
#
# processEachBiosampleXML <- function(mydata=mydata){
#          print(paste("Processing XML object with .. ",
#                      xml_length(mydata) , " Biosample ID" ,sep=" "))
#           mydata %>% xml_find_all("//SAMPLE") %>%   # Proceed per Biosample ID
#           map (~ extractSampleInfo(.x))       # Extract information per BiosampleID
#
# }
```

```
# #lapply(results, processEachBiosampleXML)       # Extract information per BiosampleID
# system.time(biosample_content <- plapply(results, processEachBiosampleXML, .parallel=6, .verbose = TR
#
# #Convert the list into dataframe
# list2df <- function(x){
#    mdf <- tbl_df(do.call(rbind,x))
#    return(mdf)
# }
#
# #Collapse all sub dataframes in one big table
# system.time(sample.info.list <- plapply(biosample_content, list2df, .parallel=6, .verbose = TRUE))
# sample.info <- sample.info.list %>%
#                  bind_rows() %>% unique()
#%>% filter(tag %in% feature_selection) %>% filter(!value=='Unknown')
#
# save(file="gutproject.sample.info.RData", sample.info)
```

```
setwd('/Users/blaise/Desktop/MetagenomicsCourse')
load(file = "gutproject.sample.info.RData")
```

Preprocess the data

```
# Retain only lineages with full species name
gutproject <- gutproject %>%
              rename(taxonomy=X.SampleID) %>%
                #mutate(taxonomy=gsub(".*?s__","",taxonomy)) %>%
                mutate(taxonomy=gsub(".*?g__","",taxonomy)) %>%
                mutate(taxonomy=gsub(";s__","_", taxonomy)) %>%
                  filter(taxonomy !="" | taxonomy=='_' ) %>%
                    filter(!grepl('_$', taxonomy))
```

```
# Merge counts of species in the same experimental sample
gutproject <- gutproject %>%
                group_by(taxonomy) %>%
                  summarize_each(funs(sum)) %>%
                    ungroup()
```

```
# Use the species name as the rowname, this replaces the default numerical name
rownames(gutproject) <- gutproject$taxonomy
```

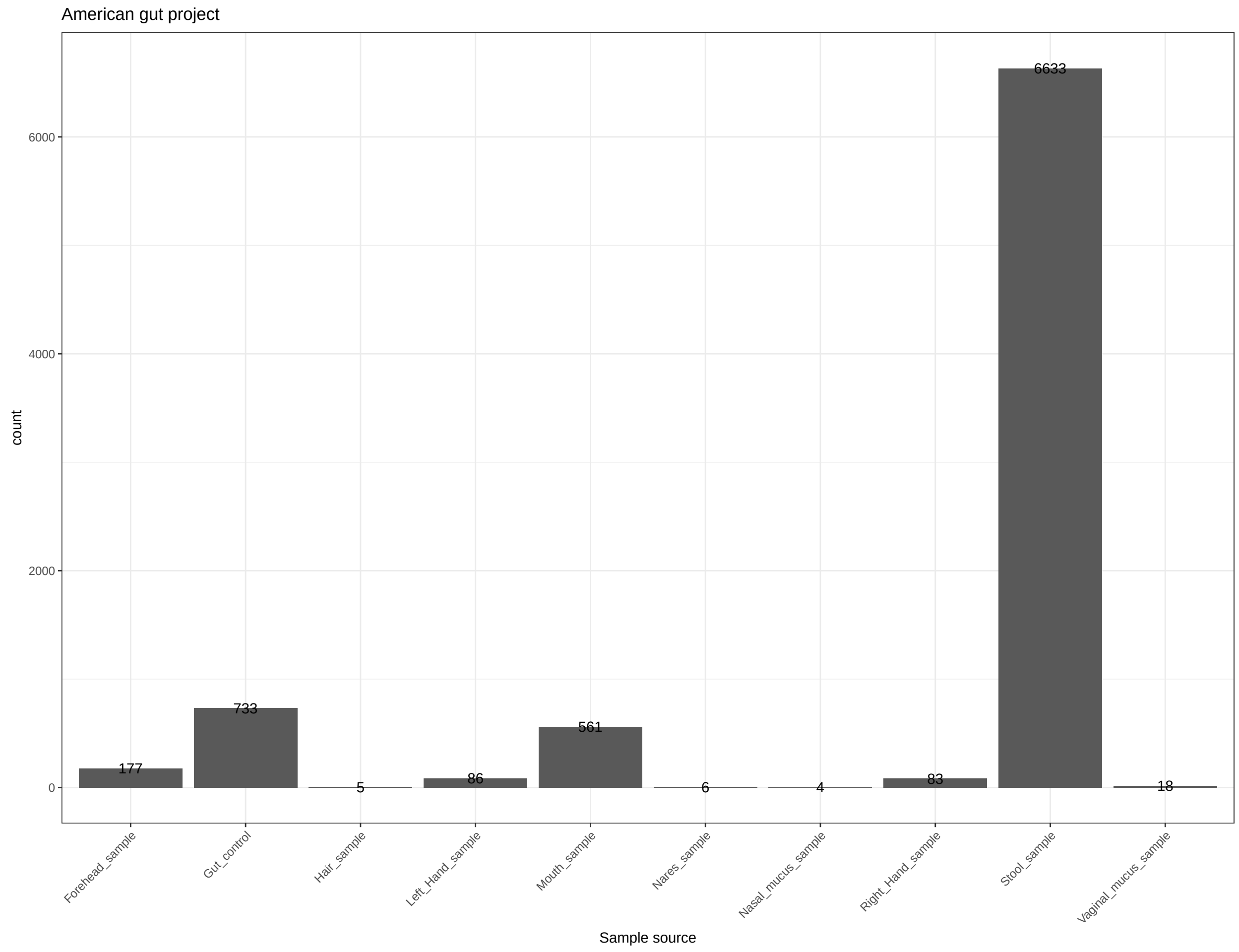
```
## Warning: Setting row names on a tibble is deprecated.
```

```
# Transform the wide format data into the long format for the purpose of appending
# descriptive information
gutproject.long  <- gutproject %>%
                      gather(Run.ID, freq, ERR1072624:ERR1160857) %>%
                        filter(taxonomy !="Root")
```

```
# Retain only the sample description and the Run id from the metadata
# Merge ENA metadata with MG metadata.
meta.MG <- meta.MG %>% mutate(Sample.ID=as.character(Sample.ID))
meta.ENA <- sample.info

meta.ENA.MG <- inner_join(meta.ENA, meta.MG, by=c("id"="Sample.ID"))
```

```
# select subset of column for downstream analysis
# Remove qualifier with unknown values

ena.meta <- meta.ENA.MG %>% select(Run.ID, tag,value) %>%
            unique() %>% filter(tag %in% sub_feature) %>%
            mutate(Run.ID=as.character(Run.ID)) %>%
            mutate(Sample.Description=paste(tag,value ,sep=":")) %>%
            select(Run.ID,Sample.Description) %>%
            filter(!grepl('Unknown', Sample.Description))  %>%
            filter(!grepl('I do not have', Sample.Description))  %>%
            unique()

mg.meta  <- meta.ENA.MG %>% select(Run.ID, Sample.Description,id) %>%
  unique() %>%
  mutate(Run.ID=as.character(Run.ID))
```

```
# What is the data made up of.

meta.strip <- mg.meta %>% select(Sample.Description, Run.ID,id) %>%
              unique() %>% group_by(Sample.Description) %>%
              mutate(count=n()) %>% ungroup() %>%
              select(Sample.Description, count)  %>% unique()

meta.strip %>% ggplot(aes(x=Sample.Description, y=count)) +
                geom_bar(stat='identity') +
                geom_text(aes(label=count))  + theme_bw() +
                      theme(axis.text.x=element_text(angle=45, hjust=1)) +
                      ggtitle("American gut project") + xlab("Sample source")
```



```
####################################################
# Merge metagenomics contingency table with the
# the detailed ENA metadata for each biosample_ID
####################################################
system.time(gutproject.ena <- inner_join(gutproject.long, ena.meta, by=c("Run.ID"="Run.ID")))
```

```
##    user  system elapsed
##   2.950   0.875   4.222
```

```
system.time(gutproject.mg <- inner_join(gutproject.long, mg.meta, by=c("Run.ID"="Run.ID")))
```

```
##    user  system elapsed
##   0.852   0.221   1.080
```

Merge both annotations_

```
system.time(gutproject.ena <- gutproject.ena %>%
              group_by(taxonomy,Sample.Description) %>%
              mutate(freq=sum(freq)) %>%
              ungroup() %>%
              select(taxonomy,Sample.Description, freq) %>%
              unique())
```

```
##    user  system elapsed
##  42.725   1.498  45.155
```

```
system.time(gutproject.mg <- gutproject.mg %>% select(-Run.ID, -id) %>%
              group_by(taxonomy,Sample.Description) %>%
              mutate(freq=sum(freq)) %>%
              ungroup() %>%
              select(taxonomy,Sample.Description, freq) %>%
              unique())
```

```
##    user  system elapsed
##   4.613   0.131   4.791
```

```
gutproject.df <- bind_rows(gutproject.ena, gutproject.mg)
```

–Transform the long format to wide format for the purpose of Mutivariate analysis___

```
gutproject.df <- gutproject.df %>%
                     spread(Sample.Description, freq, fill=0) %>%
                       as.data.frame()
gutproject.mat <- gutproject.df %>% select(-c(1)) %>% as.matrix()
rownames(gutproject.mat) <- gsub("\\[|\\]","", perl=TRUE, gutproject.df$taxonomy)
```

```
gutproject.mat <-  gutproject.mat[!rownames(gutproject.mat)=="",]
#Remove rows that sums to 0
gutproject.mat <- gutproject.mat[which(rowSums(gutproject.mat)!=0),]
#Remove columns that sums to 0
gutproject.mat <- gutproject.mat[,which(colSums(gutproject.mat)!=0)]
```

Is there any association between the species and the data source?

```
# Perform a chi-square test of independence, this evaluates whether there is a significant
# dependence between species and biological samples

chisq <- chisq.test(gutproject.mat)
```

```
## Warning in chisq.test(gutproject.mat): Chi-squared approximation may be
## incorrect
```

```
chisq
```

```
##
##  Pearson's Chi-squared test
##
## data:  gutproject.mat
## X-squared = 162350000, df = 57218, p-value < 2.2e-16
```

```
# The species and biological samples are statistically significantly associated (p-value=0)
```

Perform Corresspondance Anlysis, CA (a generalized from of Principal Component Analysis for categorical data)

```
# We use CA to reduce the dimension of the data without loosing the most important information.
# CA is used to graphically visualize rows points and column points in a low dimensional space

gutproject.ca <- CA(gutproject.mat, graph=FALSE)
```

Explore the content of the CA output

```
summary(gutproject.ca, nb.dec = 2, nbelements = 4, ncp= 3)
```

```
##
```

```
## Call:
## CA(X = gutproject.mat, graph = FALSE)
##
## The chi square of independence between the two variables is equal to 162346345 (p-value =  0 ).
##
## Eigenvalues
##                        Dim.1  Dim.2  Dim.3  Dim.4  Dim.5  Dim.6  Dim.7
## Variance                0.11   0.06   0.03   0.02   0.02   0.01   0.01
## % of var.              33.44  18.36   8.66   7.32   6.58   3.61   3.19
## Cumulative % of var.   33.44  51.80  60.46  67.78  74.36  77.96  81.16
##                        Dim.8  Dim.9 Dim.10 Dim.11 Dim.12 Dim.13 Dim.14
## Variance                0.01   0.01   0.01   0.00   0.00   0.00   0.00
## % of var.               2.43   2.18   1.67   1.36   1.34   0.99   0.86
## Cumulative % of var.   83.59  85.77  87.44  88.80  90.15  91.14  92.01
##                       Dim.15 Dim.16 Dim.17 Dim.18 Dim.19 Dim.20 Dim.21
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.77   0.71   0.67   0.63   0.50   0.46   0.43
## Cumulative % of var.   92.77  93.49  94.16  94.79  95.29  95.75  96.18
##                       Dim.22 Dim.23 Dim.24 Dim.25 Dim.26 Dim.27 Dim.28
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.40   0.37   0.32   0.29   0.29   0.25   0.21
## Cumulative % of var.   96.58  96.95  97.26  97.55  97.84  98.09  98.29
##                       Dim.29 Dim.30 Dim.31 Dim.32 Dim.33 Dim.34 Dim.35
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.19   0.18   0.15   0.14   0.13   0.11   0.11
## Cumulative % of var.   98.49  98.67  98.82  98.96  99.09  99.19  99.30
##                       Dim.36 Dim.37 Dim.38 Dim.39 Dim.40 Dim.41 Dim.42
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.10   0.08   0.07   0.06   0.06   0.05   0.05
## Cumulative % of var.   99.40  99.49  99.55  99.61  99.67  99.72  99.77
##                       Dim.43 Dim.44 Dim.45 Dim.46 Dim.47 Dim.48 Dim.49
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.04   0.04   0.03   0.02   0.02   0.01   0.01
## Cumulative % of var.   99.81  99.85  99.88  99.91  99.93  99.94  99.95
##                       Dim.50 Dim.51 Dim.52 Dim.53 Dim.54 Dim.55 Dim.56
## Variance                0.00   0.00   0.00   0.00   0.00   0.00   0.00
## % of var.               0.01   0.01   0.01   0.01   0.00   0.00   0.00
## Cumulative % of var.   99.97  99.97  99.98  99.99  99.99  99.99 100.00
##                       Dim.57 Dim.58 Dim.59 Dim.60 Dim.61
## Variance                0.00   0.00   0.00   0.00   0.00
## % of var.               0.00   0.00   0.00   0.00   0.00
## Cumulative % of var.  100.00 100.00 100.00 100.00 100.00
##
## Rows (the 4 first)
##                                                                 Iner*1000
## Clostridium_difficile                                         |      0.00 |
## Clostridium_sordellii                                         |      0.00 |
## Eubacterium_biforme                                           |      0.33 |
## Eubacterium_cylindroides                                      |      0.08 |
##                                                             Dim.1   ctr
## Clostridium_difficile                                       -0.17  0.00
## Clostridium_sordellii                                        0.40  0.00
## Eubacterium_biforme                                         -0.14  0.08
## Eubacterium_cylindroides                                    -0.21  0.00
##                                                               cos2   Dim.2
## Clostridium_difficile                                         0.13 | -0.02
## Clostridium_sordellii                                         0.04 |  0.95
## Eubacterium_biforme                                           0.28 |  0.04
## Eubacterium_cylindroides                                      0.05 |  0.00
##                                                                ctr  cos2
## Clostridium_difficile                                         0.00  0.00 |
## Clostridium_sordellii                                         0.00  0.23 |
## Eubacterium_biforme                                           0.01  0.02 |
## Eubacterium_cylindroides                                      0.00  0.00 |
##                                                              Dim.3   ctr
## Clostridium_difficile                                         0.01  0.00
## Clostridium_sordellii                                         0.42  0.00
## Eubacterium_biforme                                           0.00  0.00
## Eubacterium_cylindroides                                      0.05  0.00
##                                                               cos2
## Clostridium_difficile                                         0.00 |
## Clostridium_sordellii                                         0.04 |
## Eubacterium_biforme                                           0.00 |
## Eubacterium_cylindroides                                      0.00 |
##
## Columns (the 4 first)
##                                                                Iner*1000
## alcohol_consumption:false                                    |      2.04 |
## alcohol_consumption:true                                     |      1.26 |
## alcohol_frequency:Daily                                      |      3.33 |
## alcohol_frequency:Never                                      |      2.04 |
##                                                              Dim.1   ctr
## alcohol_consumption:false                                     0.04  0.05
## alcohol_consumption:true                                      0.05  0.23
## alcohol_frequency:Daily                                       0.06  0.04
## alcohol_frequency:Never                                       0.04  0.05
##                                                               cos2   Dim.2
## alcohol_consumption:false                                     0.03 | -0.09
## alcohol_consumption:true                                      0.20 |  0.05
## alcohol_frequency:Daily                                       0.01 |  0.18
## alcohol_frequency:Never                                       0.03 | -0.09
##                                                                ctr  cos2
## alcohol_consumption:false                                     0.38  0.11 |
## alcohol_consumption:true                                      0.33  0.16 |
## alcohol_frequency:Daily                                       0.66  0.12 |
## alcohol_frequency:Never                                       0.38  0.11 |
##                                                              Dim.3   ctr
## alcohol_consumption:false                                    -0.07  0.49
## alcohol_consumption:true                                      0.00  0.01
## alcohol_frequency:Daily                                       0.14  0.90
## alcohol_frequency:Never                                      -0.07  0.49
##                                                               cos2
## alcohol_consumption:false                                     0.07 |
## alcohol_consumption:true                                      0.00 |
## alcohol_frequency:Daily                                       0.08 |
## alcohol_frequency:Never                                       0.07 |
# The result of the function summary() contains the chi-square statistics and 3 tables:
# Table 1- Eigenvalues, displays the variances and the percentage of vaiances retained by each
#         dimension.
# Table 2, Contains the coordinates, the contribution and the cos2 (quality of representation
#          [0-1] of the first 4 active rows variable on the dimension 1, 2 and 3
# Table 3: displays the coordinates, the contribution and the cos2 of the first 4 active column
#          variables on the dimension 1, 2 and 3
#
```

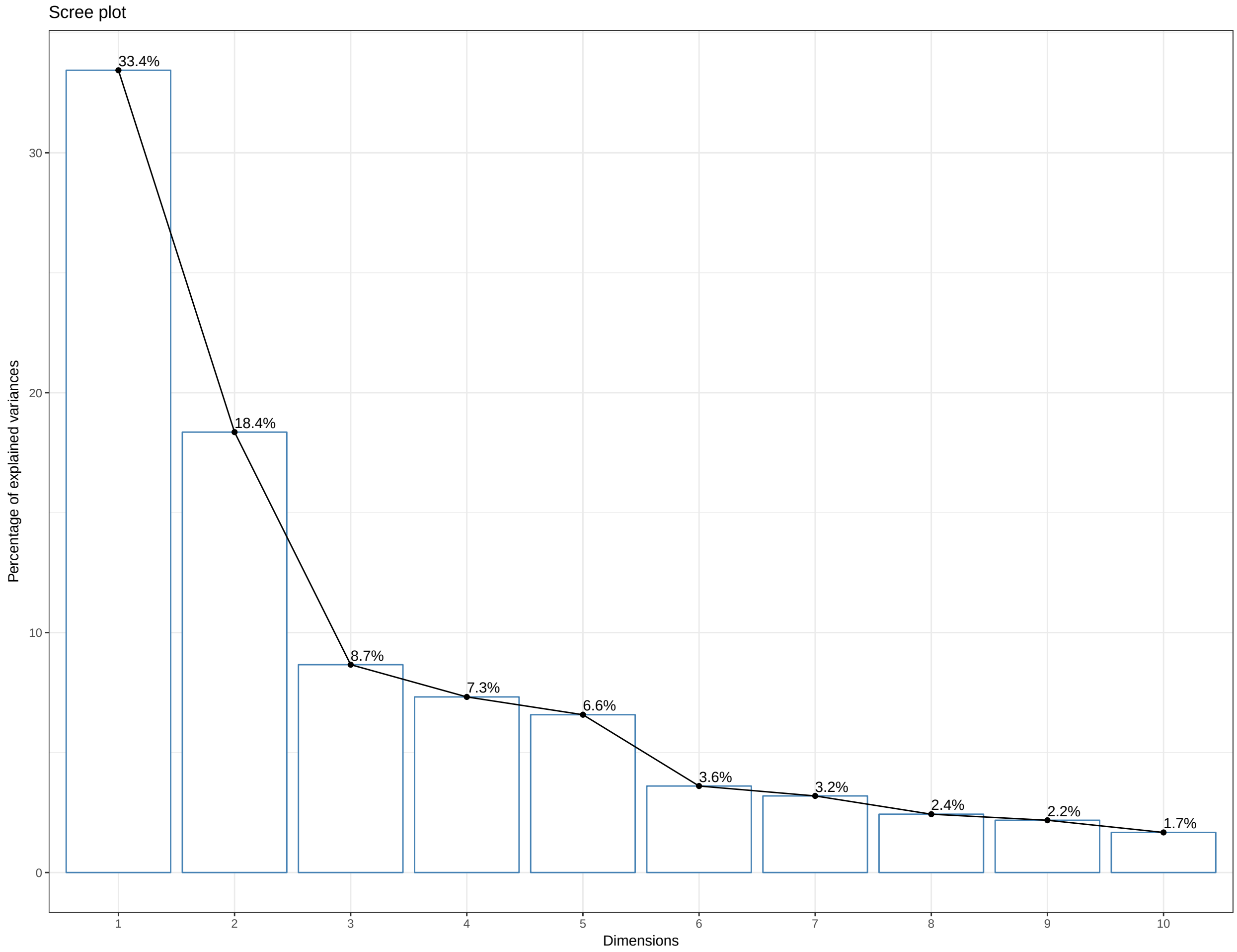
Interprete the CA result above, hint:correlation coef., chi-square statistic

```
# Correlation coefficient
eig <- get_eigenvalue(gutproject.ca)
# Eigen values is the amount of information retained by each axis.
trace <- sum(eig$eigenvalue)
cor.coef <- sqrt(trace)
cor.coef
```

```
## [1] 0.5779205
```

```
# As a rule of thumb a correlation above 0.2 can be considered important
# (Bendixen 1995, 567; Healey 2013, 289-290)
# The correlation coef. is 0.50in the American gut project dataset,
# indicating a strong association between row (species)
# and column(biological samples variables.).
# A more rigorous approach for examining the association is to use Chi-square statistics.
# The association is highly significant (p-value=0)
```

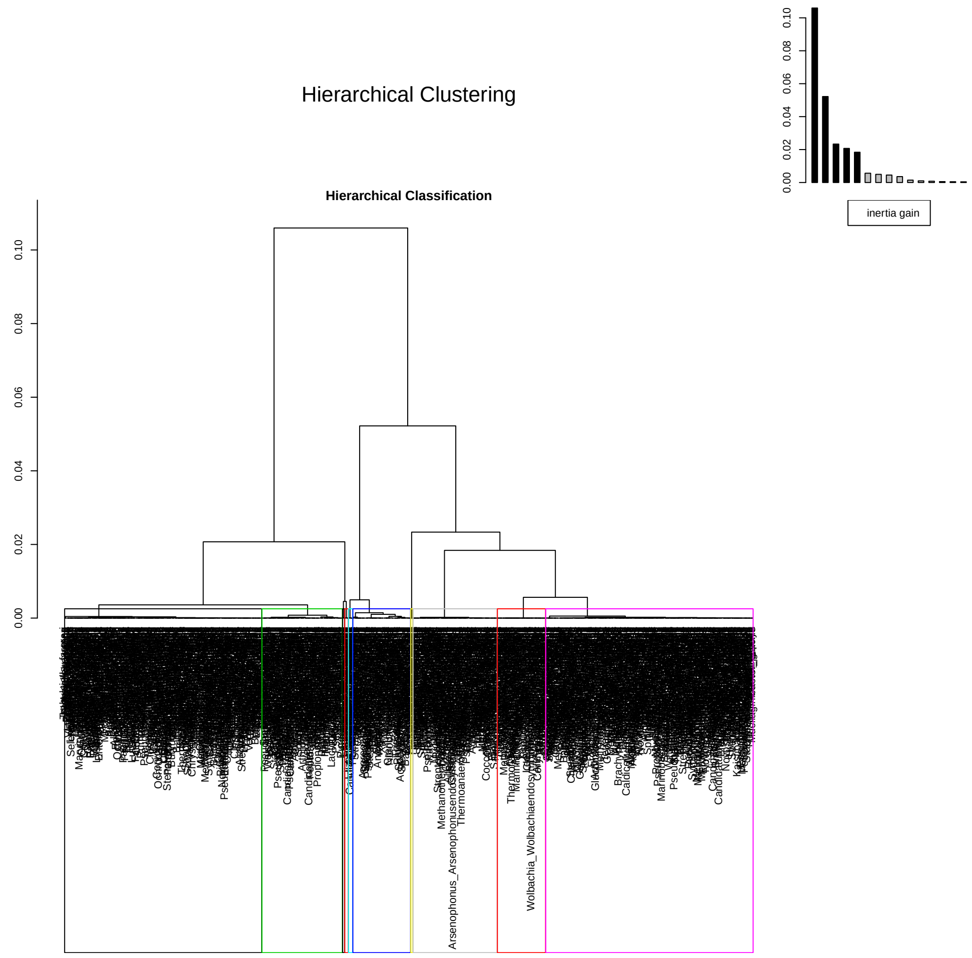
```
# Explore the percentages of inertia explained by the CA dimensions, the scree plot
fviz_screeplot(gutproject.ca, barfill='white', addlabels=TRUE) + theme_bw()
```

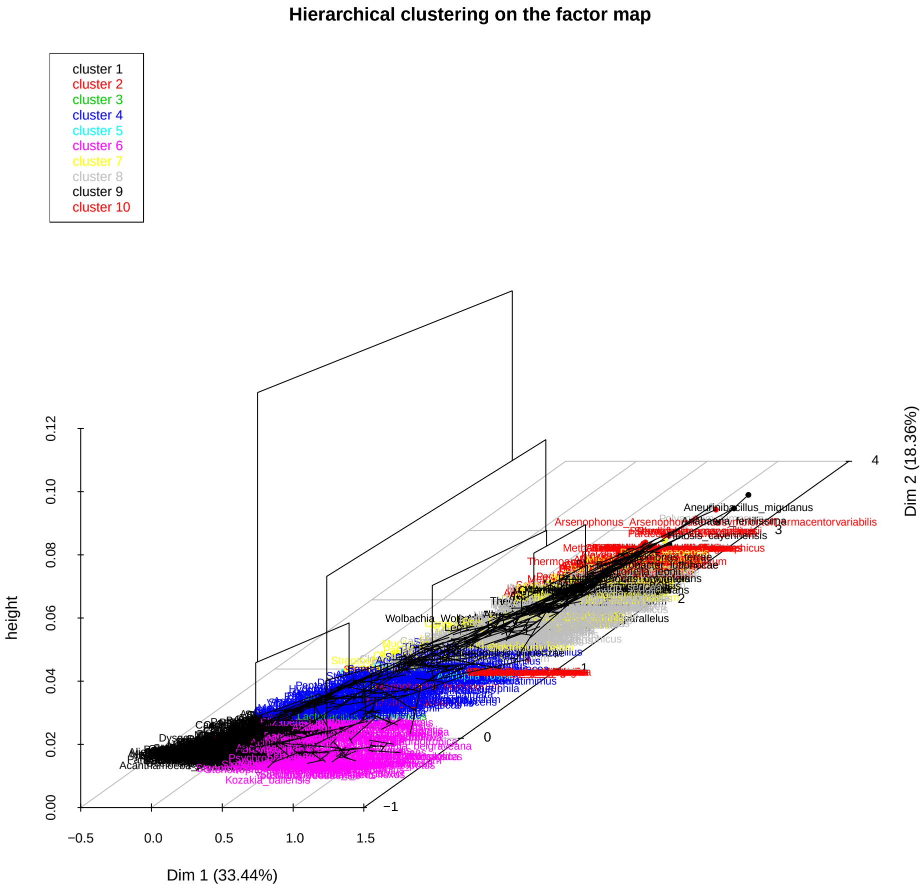


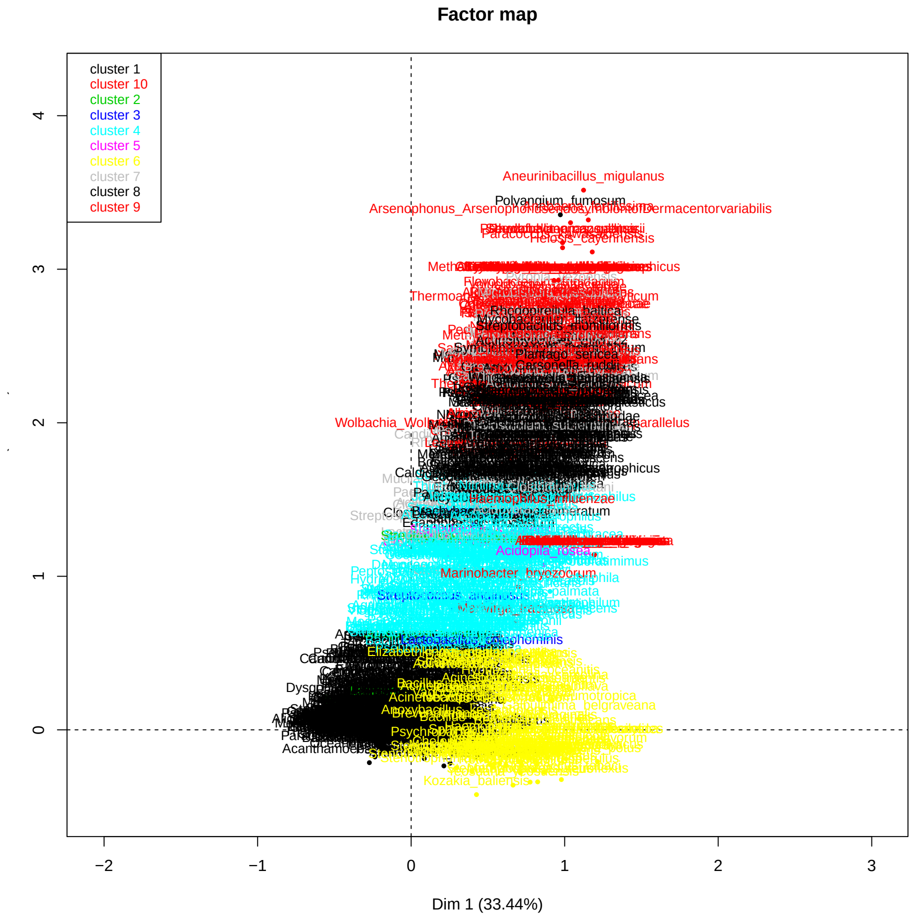
```
# The point at which the scree plot shows a bend mignt indicate an optimal dimensionality.
```

Hierarchical Clustering of principal components

```
# Compute hierarchical clustering of species of CA results
gutproject.hcpc <- HCPC(gutproject.ca, graph = TRUE , nb.clust=10, order=TRUE )
```

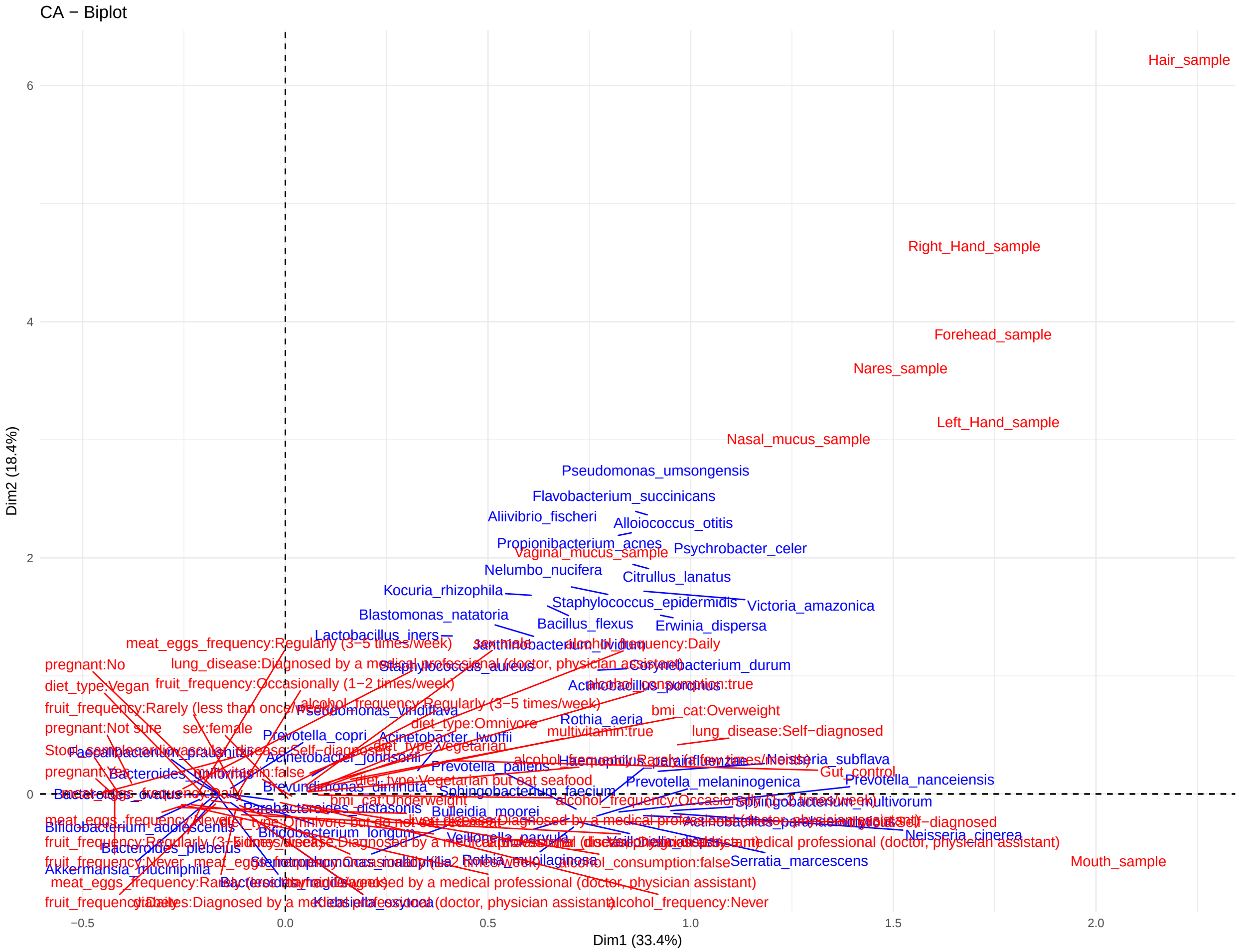

Hierarchical Clustering
Hierarchical Classification
0.00
0.02
0.04
0.06
0.08
0.10
inertia gain
Arsenophonus_Arsenophonusendo
Wolbachia_Wolbachiaendos
Methano
Thermoanae


Hierarchical clustering on the factor map
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
cluster 7
cluster 8
cluster 9
cluster 10
height
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Dim 1 (33.44%)
−0.5
0.0
0.5
1.0
1.5
Dim 2 (18.36%)
−1
0
1
2
3
4
Aneurinibacillus_migulanus
Arsenophonus_Arsenophonus
armacentorvariabilis
Wolbachia_Wol
Kozakia_baliensis
Acanthamoeba



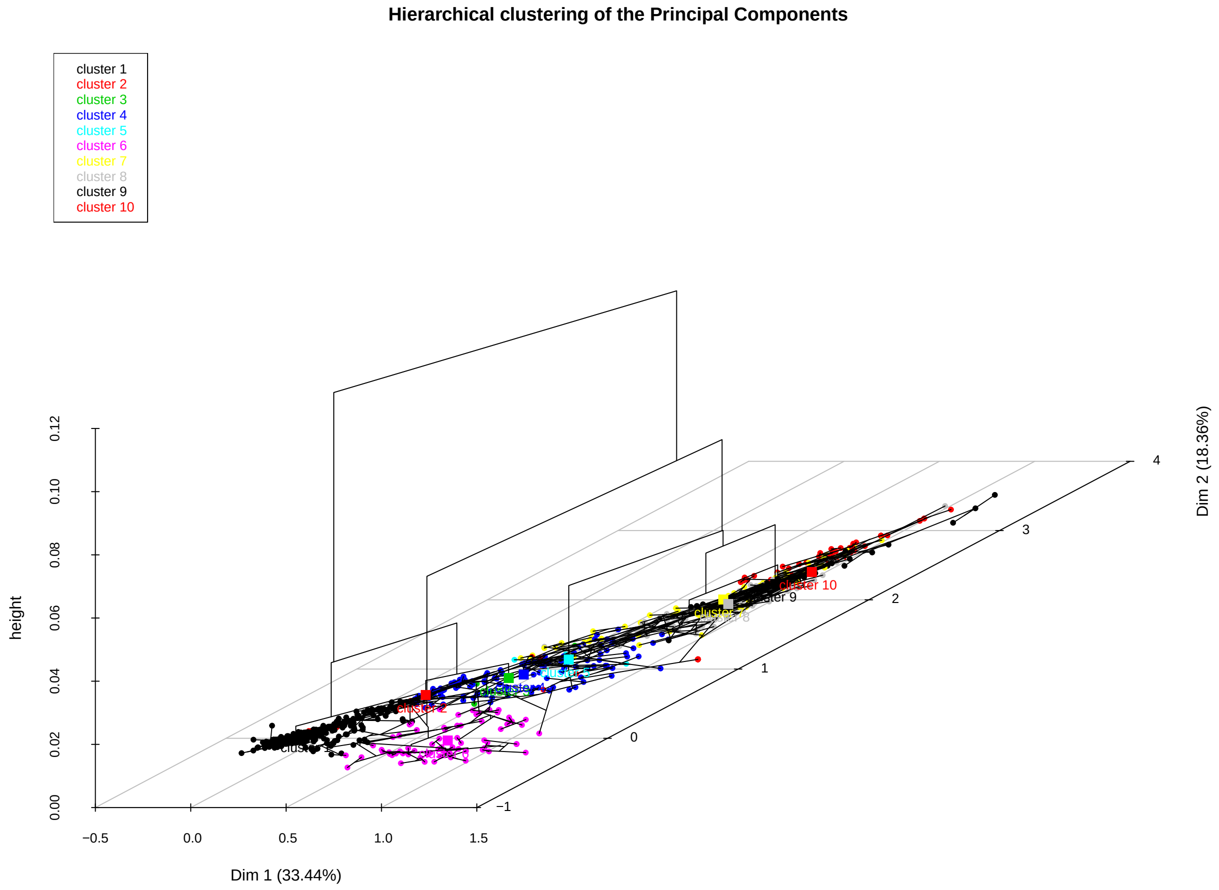
Visualize the CA results

```
# Symmetric biplot, whereby both rows (blue) and columns (red) are represented
# in the same space using the principal coordinates. Only the distance between row
# points or the distance between column points can be reliably interpreted.
# With a symmetric plot, only general statements can be drawn about the pattern.
# The inter-distance between rows and column can not be interpreted.
fviz_ca_biplot(gutproject.ca, repel=TRUE, select.row = list(contrib = 50),
               select.col = list(contrib = 50), geom="text") +
  theme_minimal()
```



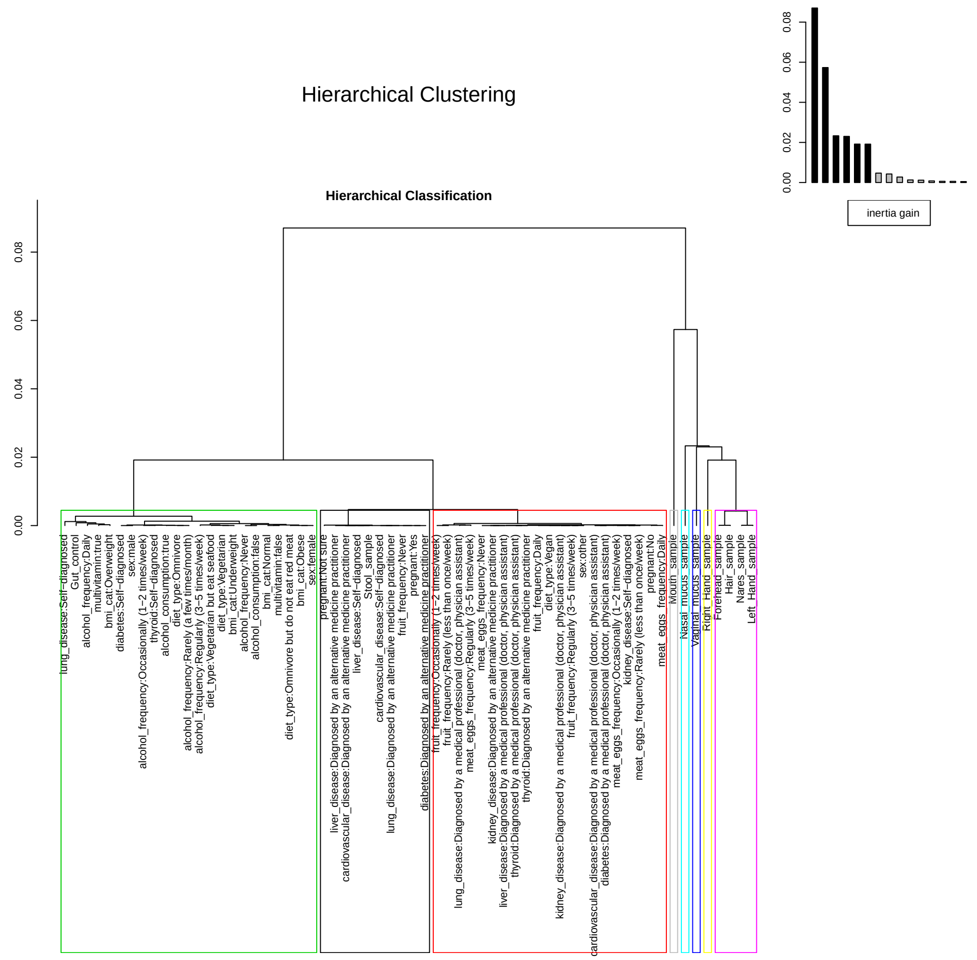
Remove labels and add cluster centers

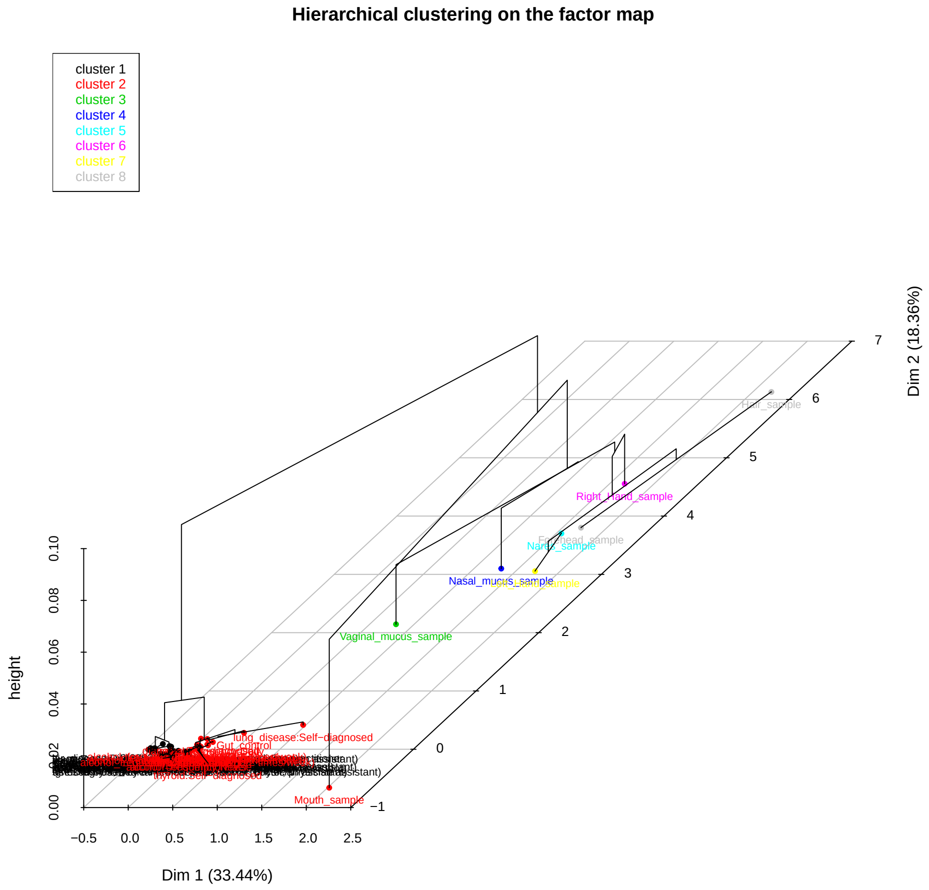
```
# A 3D map of the hierarchical clustering of the principal component.
plot(gutproject.hcpc, choice ="3D.map", draw.tree = FALSE,
     ind.names = FALSE, centers.plot = TRUE,
     title='Hierarchical clustering of the Principal Components')
```

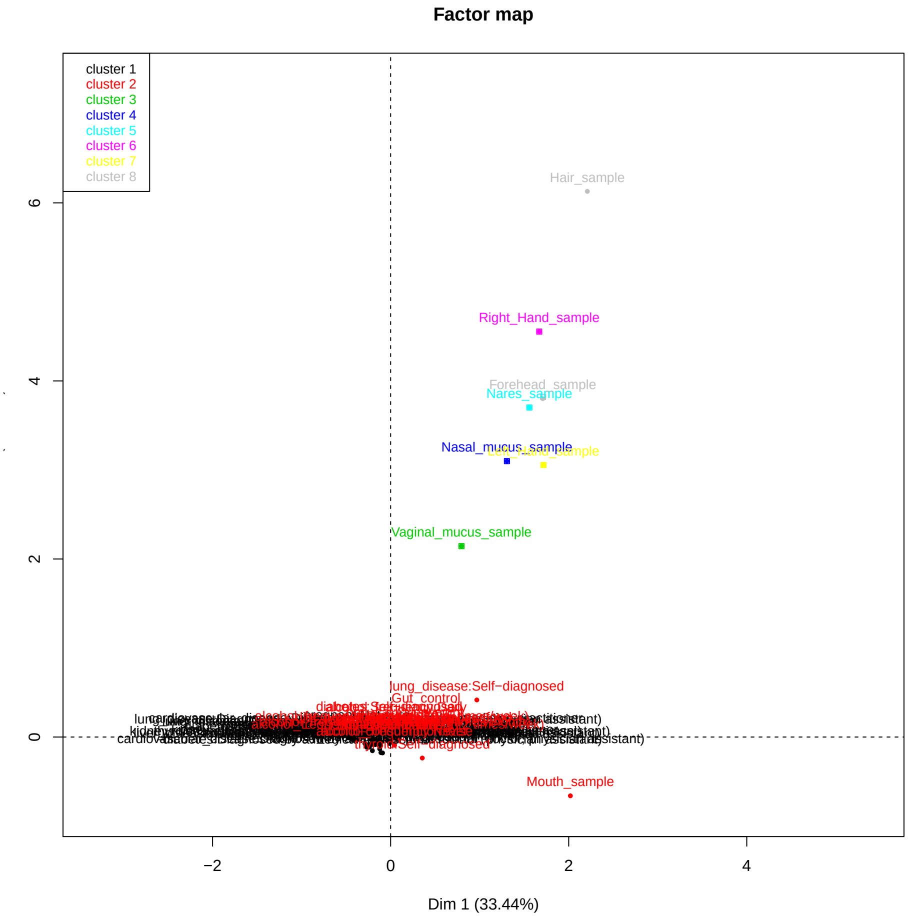


Hierarchical Clustering of principal component.

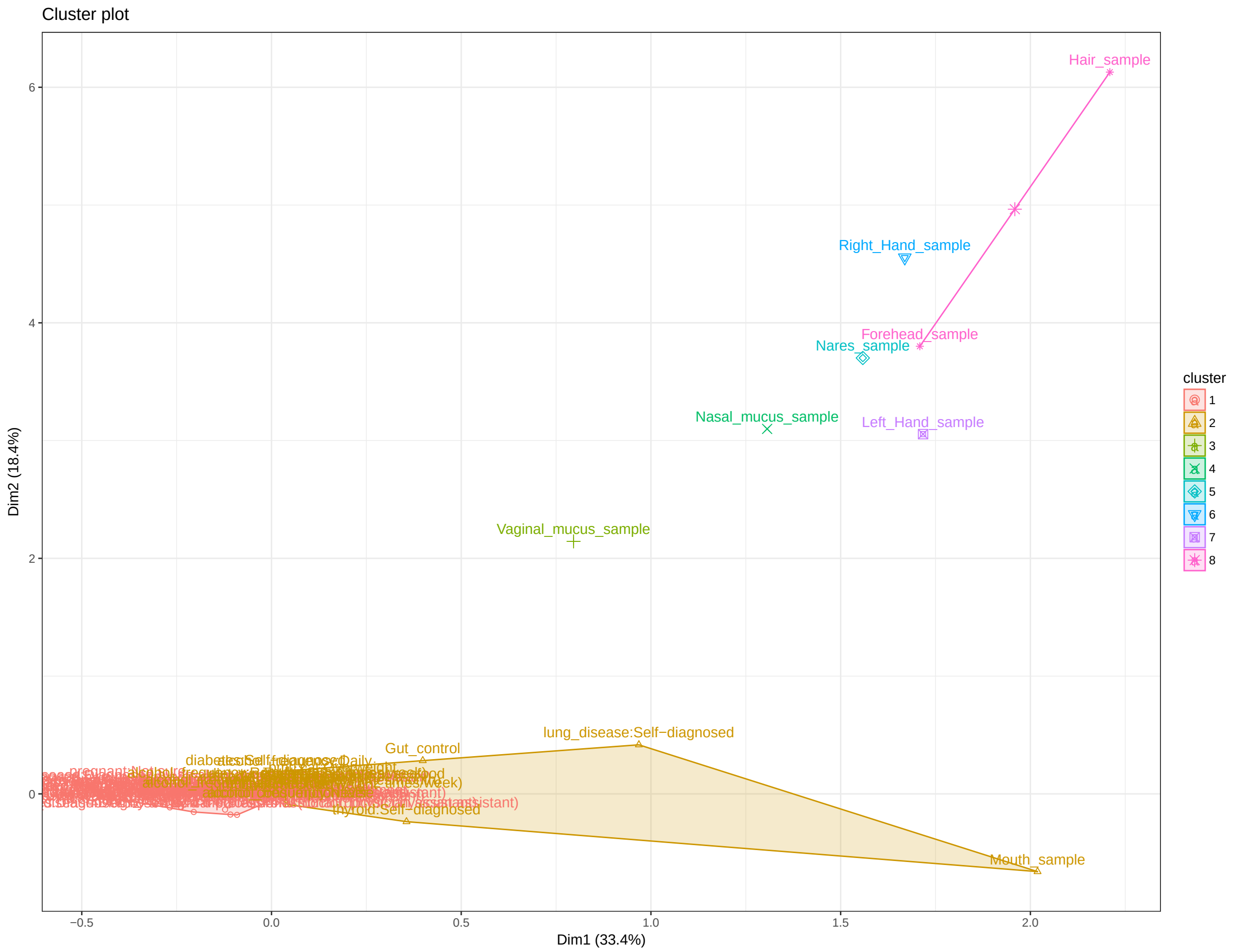
```
# Compute hierarchical clustering of biological samples of CA results
res.hcpc <- HCPC(gutproject.ca, cluster.CA = "columns",
                     graph = TRUE , nb.clust=8, order=TRUE )
```


Hierarchical Clustering
Hierarchical Classification
0.00
0.02
0.04
0.06
0.08
lung_disease:Self-diagnosed
Gut_control
alcohol_frequency:Daily
multivitamin:true
bmi_cat:Overweight
diabetes:Self-diagnosed
sex:male
alcohol_frequency:Occasionally (1−2 times/week)
thyroid:Self-diagnosed
alcohol_consumption:true
diet_type:Omnivore
alcohol_frequency:Rarely (a few times/month)
alcohol_frequency:Regularly (3−5 times/week)
diet_type:Vegetarian but eat seafood
diet_type:Vegetarian
bmi_cat:Underweight
alcohol_frequency:Never
alcohol_consumption:false
bmi_cat:Normal
multivitamin:false
diet_type:Omnivore but do not eat red meat
bmi_cat:Obese
sex:female
pregnant:Not sure
liver_disease:Diagnosed by an alternative medicine practitioner
cardiovascular_disease:Diagnosed by an alternative medicine practitioner
liver_disease:Self-diagnosed
Stool_sample
cardiovascular_disease:Self-diagnosed
lung_disease:Diagnosed by an alternative medicine practitioner
fruit_frequency:Never
pregnant:Yes
diabetes:Diagnosed by an alternative medicine practitioner
fruit_frequency:Occasionally (1−2 times/week)
fruit_frequency:Rarely (less than once/week)
lung_disease:Diagnosed by a medical professional (doctor, physician assistant)
meat_eggs_frequency:Regularly (3−5 times/week)
meat_eggs_frequency:Never
kidney_disease:Diagnosed by an alternative medicine practitioner
liver_disease:Diagnosed by a medical professional (doctor, physician assistant)
thyroid:Diagnosed by a medical professional (doctor, physician assistant)
thyroid:Diagnosed by an alternative medicine practitioner
fruit_frequency:Daily
diet_type:Vegan
kidney_disease:Diagnosed by a medical professional (doctor, physician assistant)
fruit_frequency:Regularly (3−5 times/week)
sex:other
cardiovascular_disease:Diagnosed by a medical professional (doctor, physician assistant)
diabetes:Diagnosed by a medical professional (doctor, physician assistant)
meat_eggs_frequency:Occasionally (1−2 times/week)
kidney_disease:Self-diagnosed
meat_eggs_frequency:Rarely (less than once/week)
pregnant:No
meat_eggs_frequency:Daily
Mouth_sample
Nasal_mucus_sample
Vaginal_mucus_sample
Right_Hand_sample
Forehead_sample
Hair_sample
Nares_sample
Left_Hand_sample
0.00
0.02
0.04
0.06
0.08
inertia gain

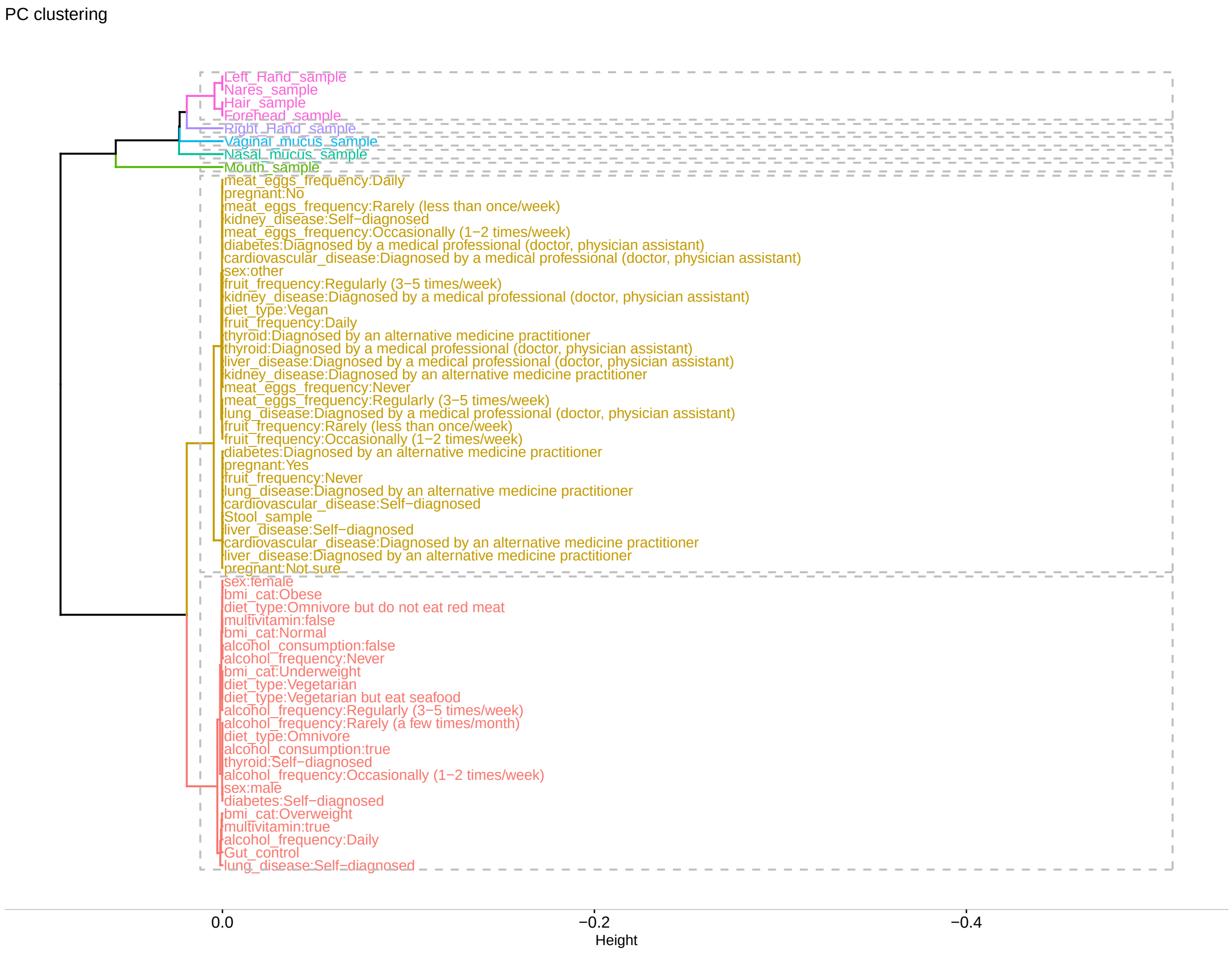

Hierarchical clustering on the factor map
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
cluster 7
cluster 8
height
0.00
0.02
0.04
0.06
0.08
0.10
Dim 1 (33.44%)
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Dim 2 (18.36%)
−1
0
1
2
3
4
5
6
7
Hair_sample
Right_Hand_sample
Forehead_sample
Nares_sample
Nasal_mucus_sample
Left_Hand_sample
Vaginal_mucus_sample
lung_disease:Self-diagnosed
Gut_control
thyroid:Self-diagnosed
Mouth_sample



```
# Biological sample cluster in the dataset
fviz_cluster(res.hcpc ) + theme_bw()
```

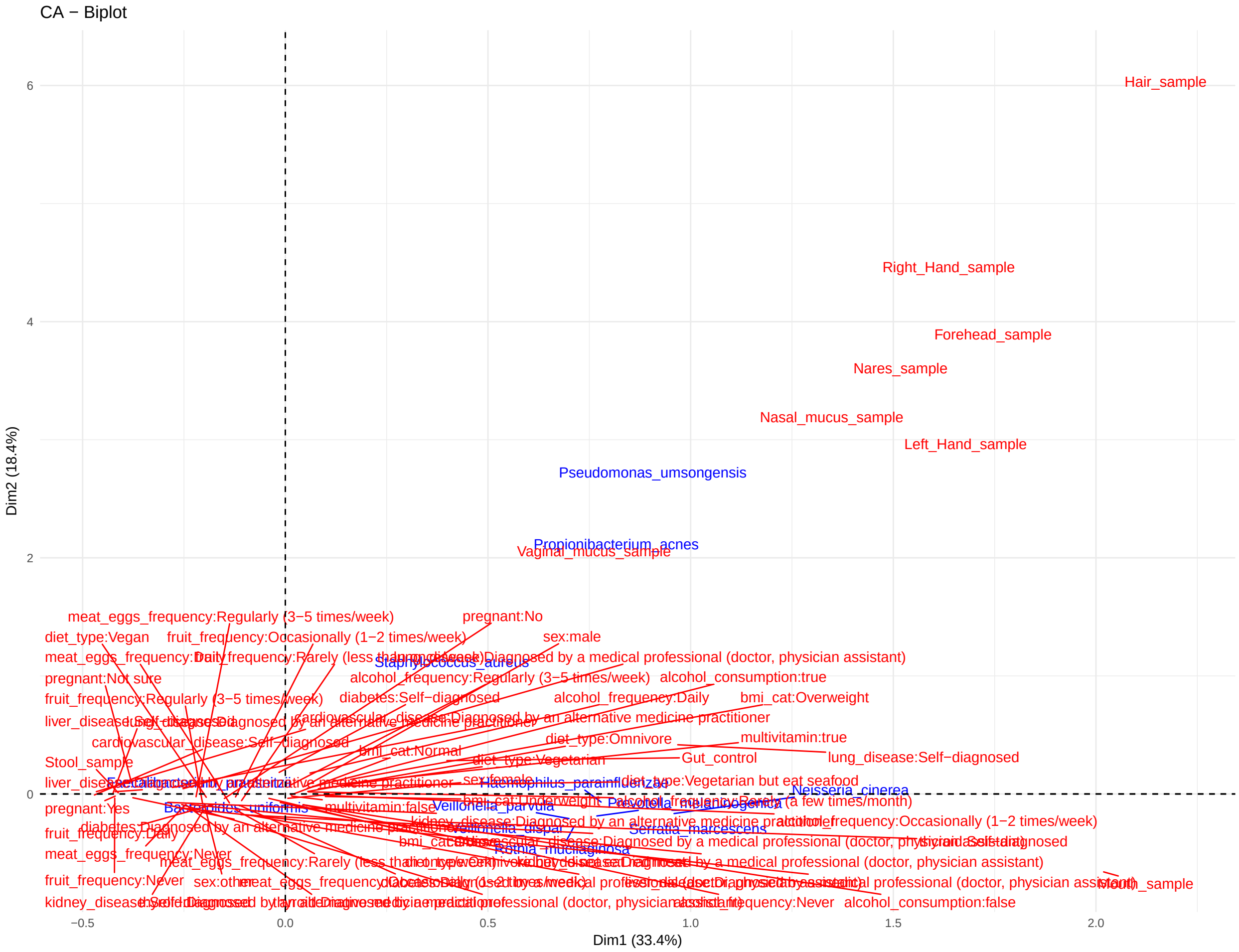


```
# Clustering of sample qualifiers....s
fviz_dend(res.hcpc, horiz=T, rec_fill=TRUE, rect = TRUE,
          type="rectangle", k=5, sub=NULL, main="PC clustering") +
     theme( axis.line.x  = element_line(colour = "grey", size = .3, linetype = "solid"))
```



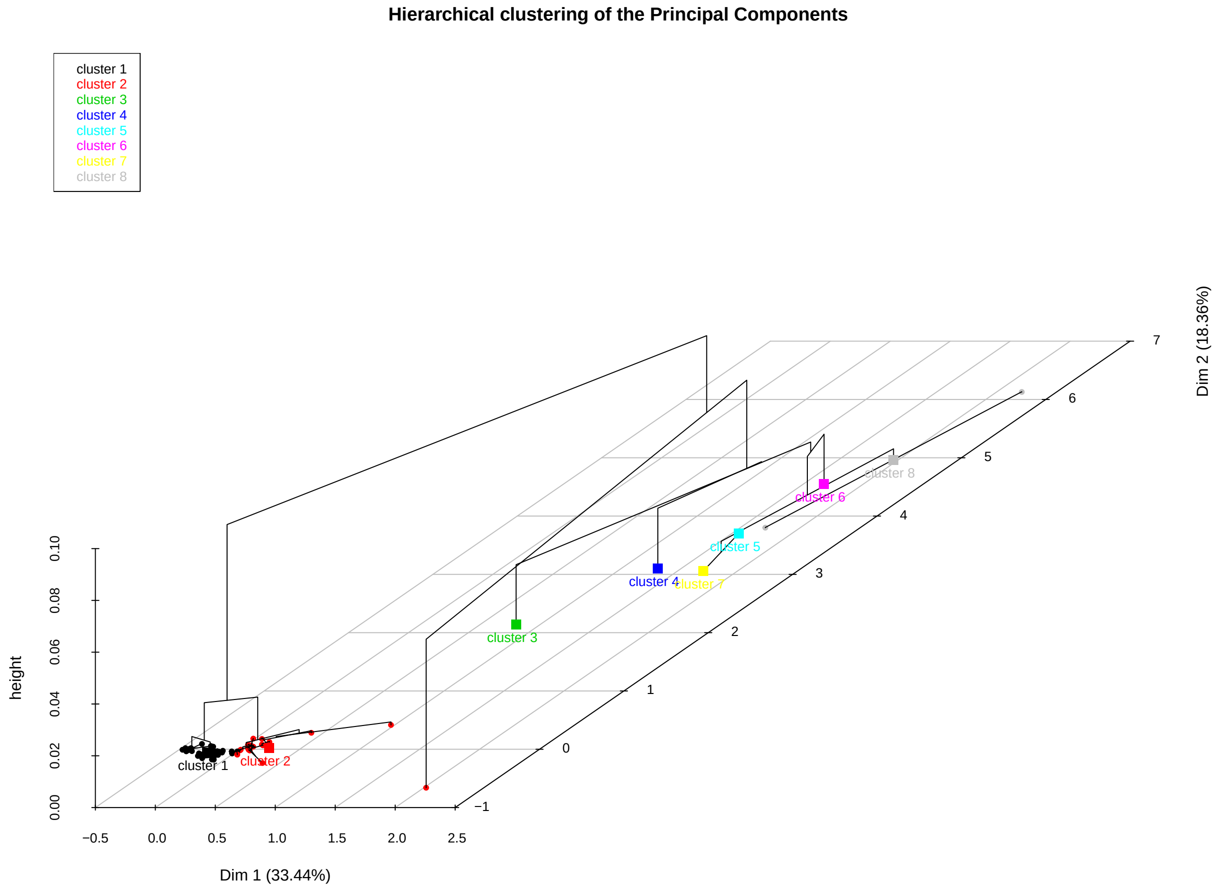
```
# transpose the row and column in the gut data and perform
# a new correspondance analysis
#gutproject.ca <- CA(t(gutproject.mat), ncp=4)
```

```
# Symmetric biplot
fviz_ca_biplot(gutproject.ca, repel=TRUE, select.row = list(contrib = 12),
               select.col = list(contrib = 95), geom="text") +
  theme_minimal()
```



Remove labels and add cluster centers

```
# 3d hierarchical clustering of the principal components.
plot(res.hcpc, choice ="3D.map", draw.tree = FALSE,
     ind.names = FALSE, centers.plot = TRUE,
     title='Hierarchical clustering of the Principal Components')
```



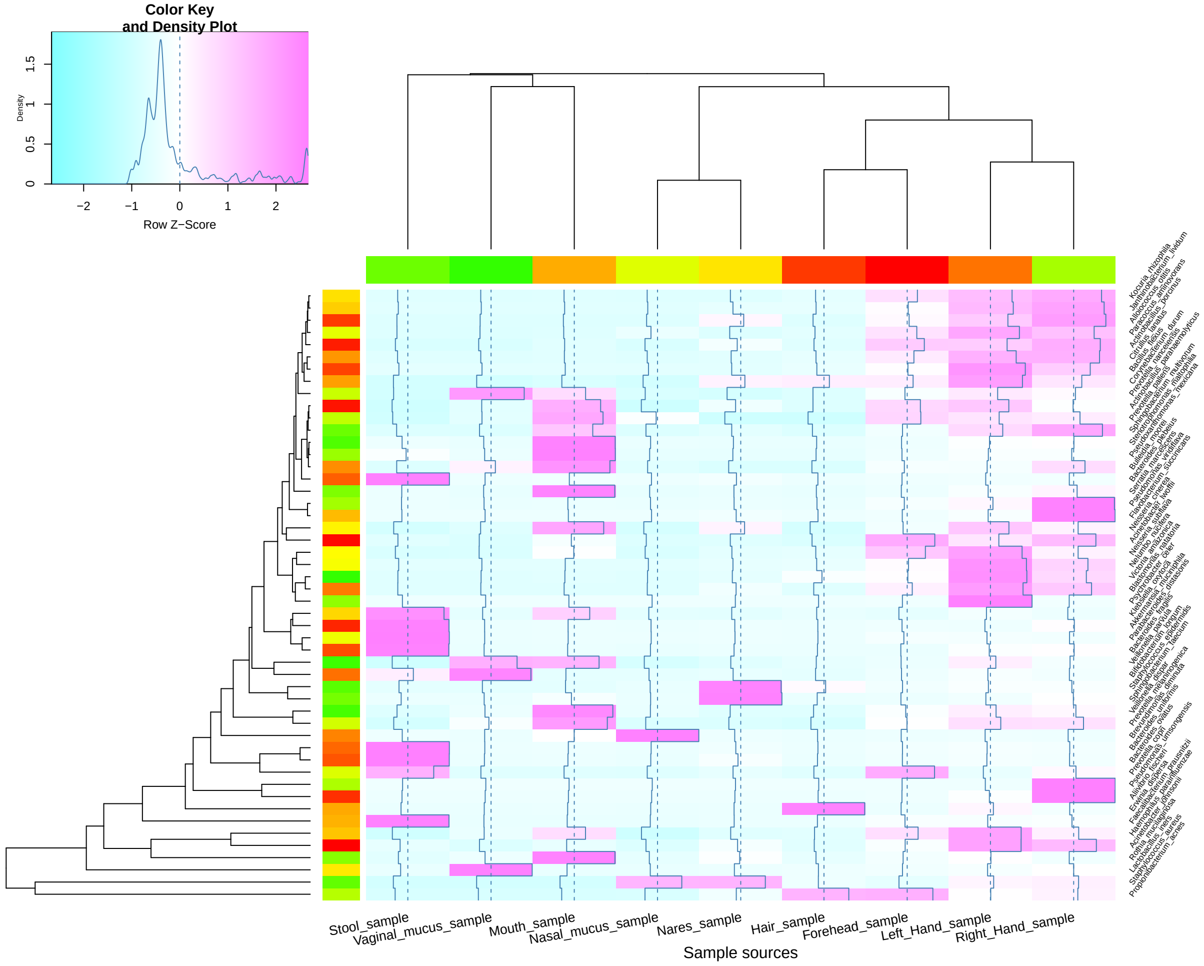
```
z_score <- function (X){
  pop_sd <- sd(X)*sqrt((length(X)-1)/(length(X)))
  pop_mean <- mean(X)
  z_score_val <- (X - pop_mean)/pop_sd
}
##################################################
# Calculate norm of vector for normalization....
##################################################
norm_vec <- function(x) x/sqrt(sum(x^2))
```

Association between species and sample source and qualifiers

```
gutproject.mat.ena <- gutproject.mat[,!grepl('sample',colnames(gutproject.mat))]
gutproject.mat.mg <- gutproject.mat[,grepl('sample',colnames(gutproject.mat))]
gutproject.mat.merge <- gutproject.mat
```

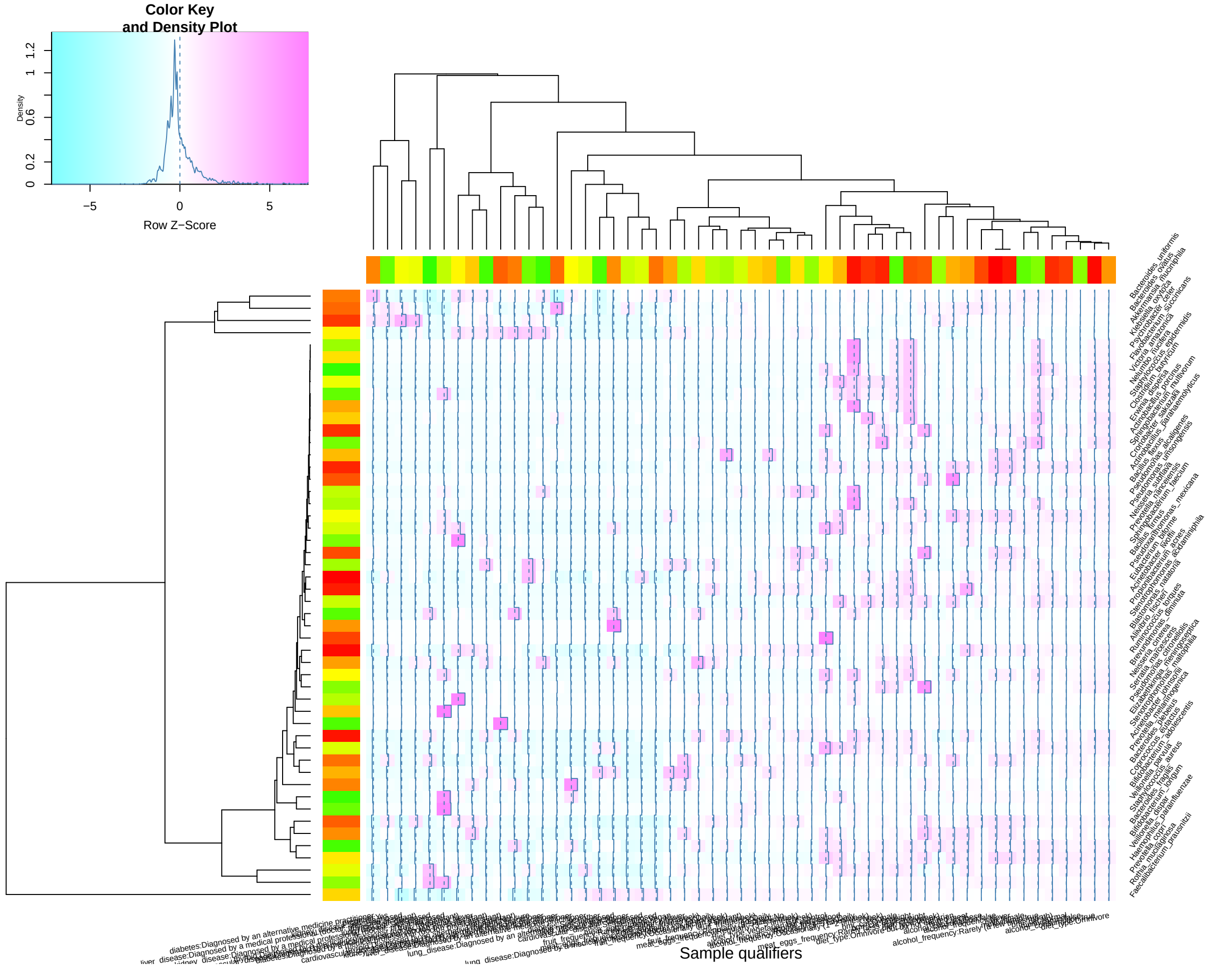
Association between species and sample source (MG portal metadata)

```
gutproject.mat.mg <-  gutproject.mat.mg[!rownames(gutproject.mat.mg)=="",]
#Remove rows that sums to 0
gutproject.mat.mg <- gutproject.mat.mg[which(rowSums(gutproject.mat.mg)!=0),]
#Remove columns that sums to 0
gutproject.mat.mg <- gutproject.mat.mg[,which(colSums(gutproject.mat.mg)!=0)]
gutproject.ca <- CA(gutproject.mat.mg, graph = FALSE)
gutproject.ca.ctrb  <- fviz_ca_row(gutproject.ca, alpha.row="contrib",
                                   select.row=list(contrib=50))
bestref.ctrb <- gutproject.ca.ctrb$data[,"name"]
gutproject.mat.sub <- gutproject.mat.mg[rownames(gutproject.mat.mg) %in% bestref.ctrb, ]
gutproject.mat.norm <- apply(gutproject.mat.sub, MARGIN = 2,
                             FUN = function(X) (norm_vec(X)))



#####################################################
#            .....Heatmap MG metadata ...
#####################################################
rc <- rainbow(nrow(gutproject.mat.norm), start=0, end=.3)
cc <- rainbow(ncol(gutproject.mat.norm), start=0, end=.3)
heatmap.2(gutproject.mat.norm, col=cm.colors(255), scale="row",
                        RowSideColors=rc, ColSideColors=cc, margin=c(5, 10),
                        xlab="Sample sources", ylab= "",
                        main="",
                        tracecol="steelblue", density="density",
                        srtRow=55, adjRow=c(0, 1), srtCol=10, adjCol=c(1,1)
)
```



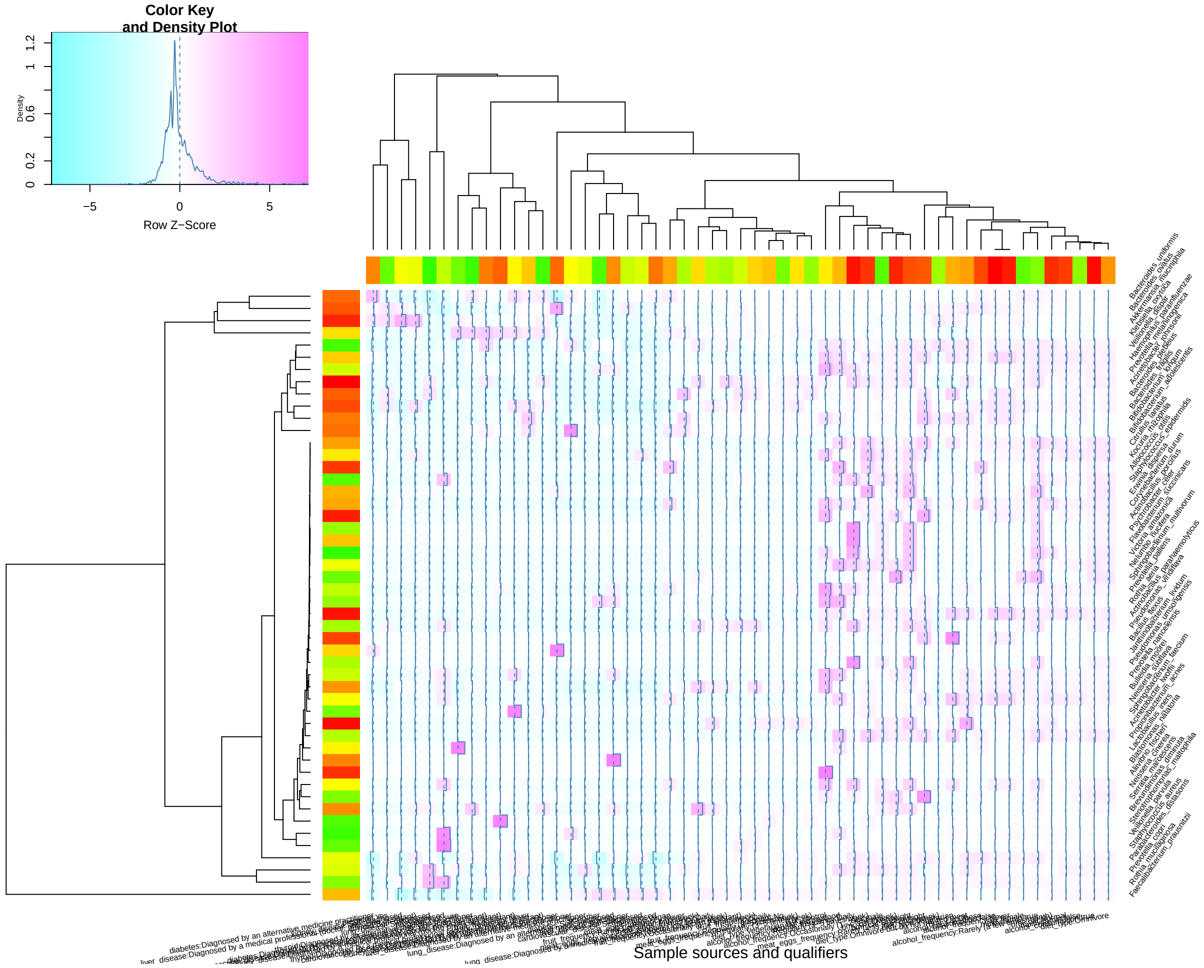
Association between species and sample qualifiers (ENA portal metadata)

```
gutproject.mat.ena <-  gutproject.mat.ena[!rownames(gutproject.mat.ena)=="",]
#Remove rows that sums to 0
gutproject.mat.ena <- gutproject.mat.ena[which(rowSums(gutproject.mat.ena)!=0),]
#Remove columns that sums to 0
gutproject.mat.ena <- gutproject.mat.ena[,which(colSums(gutproject.mat.ena)!=0)]

gutproject.ca <- CA(gutproject.mat.ena, graph = FALSE)
gutproject.ca.ctrb  <- fviz_ca_row(gutproject.ca, alpha.row="contrib",
                                   select.row=list(contrib=50))
bestref.ctrb <- gutproject.ca.ctrb$data[,"name"]
gutproject.mat.sub <- gutproject.mat.ena[rownames(gutproject.mat.ena) %in% bestref.ctrb, ]
#corpmat.ctrb <- rcorr(t(mydf.mat.ctrb))
gutproject.mat.norm <- apply(gutproject.mat.sub, MARGIN = 2,
                             FUN = function(X) (norm_vec(X)))

####################################################
#             .....Heatmap ENA metadata ...
####################################################
rc <- rainbow(nrow(gutproject.mat.norm), start=0, end=.3)
cc <- rainbow(ncol(gutproject.mat.norm), start=0, end=.3)
heatmap.2(gutproject.mat.norm, col=cm.colors(255), scale="row",
                       RowSideColors=rc, ColSideColors=cc, margin=c(5, 10),
                       xlab="Sample qualifiers", ylab= "",
                       main="",
                       tracecol="steelblue", density="density",
                       srtRow=55, adjRow=c(0, 1), srtCol=10, adjCol=c(1,1)
)
```



Association between species and sample qualifers and source (MG + ENA portal metadata)

```
gutproject.mat.merge <-  gutproject.mat.merge[!rownames(gutproject.mat.merge)=="",]
#Remove rows that sums to 0
gutproject.mat.merge <- gutproject.mat.merge[which(rowSums(gutproject.mat.merge)!=0),]
#Remove columns that sums to 0
gutproject.mat.merge <- gutproject.mat.merge[,which(colSums(gutproject.mat.merge)!=0)]

gutproject.ca <- CA(gutproject.mat.merge, graph = FALSE)
gutproject.ca.ctrb  <- fviz_ca_row(gutproject.ca, alpha.row="contrib",
                                   select.row=list(contrib=50))
bestref.ctrb <- gutproject.ca.ctrb$data[,"name"]
gutproject.mat.sub <- gutproject.mat.ena[rownames(gutproject.mat.ena) %in% bestref.ctrb, ]
#corpmat.ctrb <- rcorr(t(mydf.mat.ctrb))
gutproject.mat.norm <- apply(gutproject.mat.sub,
                             MARGIN = 2, FUN = function(X) (norm_vec(X)))

######################################################
#             .....Heatmap Merge metadata ...
######################################################
rc <- rainbow(nrow(gutproject.mat.norm), start=0, end=.3)
cc <- rainbow(ncol(gutproject.mat.norm), start=0, end=.3)
heatmap.2(gutproject.mat.norm, col=cm.colors(255), scale="row",
                     RowSideColors=rc, ColSideColors=cc, margin=c(5, 10),
                       xlab="Sample sources and qualifiers", ylab= "",
                       main="",
                       tracecol="steelblue", density="density",
                       srtRow=55, adjRow=c(0, 1), srtCol=10, adjCol=c(1,1)
)
```



APPENDIX

To reproduce this hands on session, your R sessionInfo() must be identical to the following:

```
sessionInfo()
```

```
## R version 3.3.2 (2016-10-31)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS Sierra 10.12.5
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
```

```
##  [1] webshot_0.4.0.9000  gplots_3.0.1        surveillance_1.13.1
##  [4] polyCub_0.6.0       xtable_1.8-2        sp_1.2-4
##  [7] purrr_0.2.2         xmlview_0.4.7       xml2_1.1.1
## [10] XML_3.98-1.5        FactoMineR_1.35     factoextra_1.0.4
## [13] tidyr_0.6.1         dplyr_0.5.0         ggplot2_2.2.1
## [16] ade4_1.7-5          stringr_1.2.0
##
## loaded via a namespace (and not attached):
##  [1] viridis_0.4.0        httr_1.2.1           viridisLite_0.2.0
##  [4] gtools_3.5.0         assertthat_0.2.0     stats4_3.3.2
##  [7] robustbase_0.92-7    yaml_2.1.14          ggrepel_0.6.5
## [10] backports_1.0.5      lattice_0.20-34      digest_0.6.12
## [13] polyclip_1.5-6       colorspace_1.3-2     htmltools_0.3.6
## [16] Matrix_1.2-7.1       plyr_1.8.4           devtools_1.12.0
## [19] mvtnorm_1.0-5        scales_0.4.1         gdata_2.17.0
## [22] whisker_0.3-2        tensor_1.5           git2r_0.18.0
## [25] tibble_1.3.1         mgcv_1.8-15          ggpubr_0.1.1
## [28] withr_1.0.2          nnet_7.3-12          lazyeval_0.2.0
## [31] magrittr_1.5         deldir_0.1-12        mclust_5.2.2
## [34] memoise_1.0.0        evaluate_0.10        nlme_3.1-128
## [37] MASS_7.3-45          class_7.3-14         tools_3.3.2
## [40] trimcluster_0.1-2    kernlab_0.9-25       munsell_0.4.3
## [43] cluster_2.0.5        fpc_2.1-10           flashClust_1.01-2
## [46] caTools_1.17.1       rlang_0.1.1          htmlwidgets_0.8
## [49] goftest_1.0-4        leaps_3.0            bitops_1.0-6
## [52] labeling_0.3         rmarkdown_1.3        gtable_0.2.0
## [55] abind_1.4-5          flexmix_2.3-13       DBI_0.5-1
## [58] curl_2.3             R6_2.2.0             gridExtra_2.2.1
## [61] knitr_1.15.1         prabclus_2.2-6       rprojroot_1.2
## [64] KernSmooth_2.23-15   dendextend_1.5.2     modeltools_0.2-21
## [67] stringi_1.1.2        spatstat_1.49-0      Rcpp_0.12.10
## [70] rpart_4.1-10         DEoptimR_1.0-8       diptest_0.75-7
## [73] scatterplot3d_0.3-38
```