pheatmap_tutorial.R

Adrian

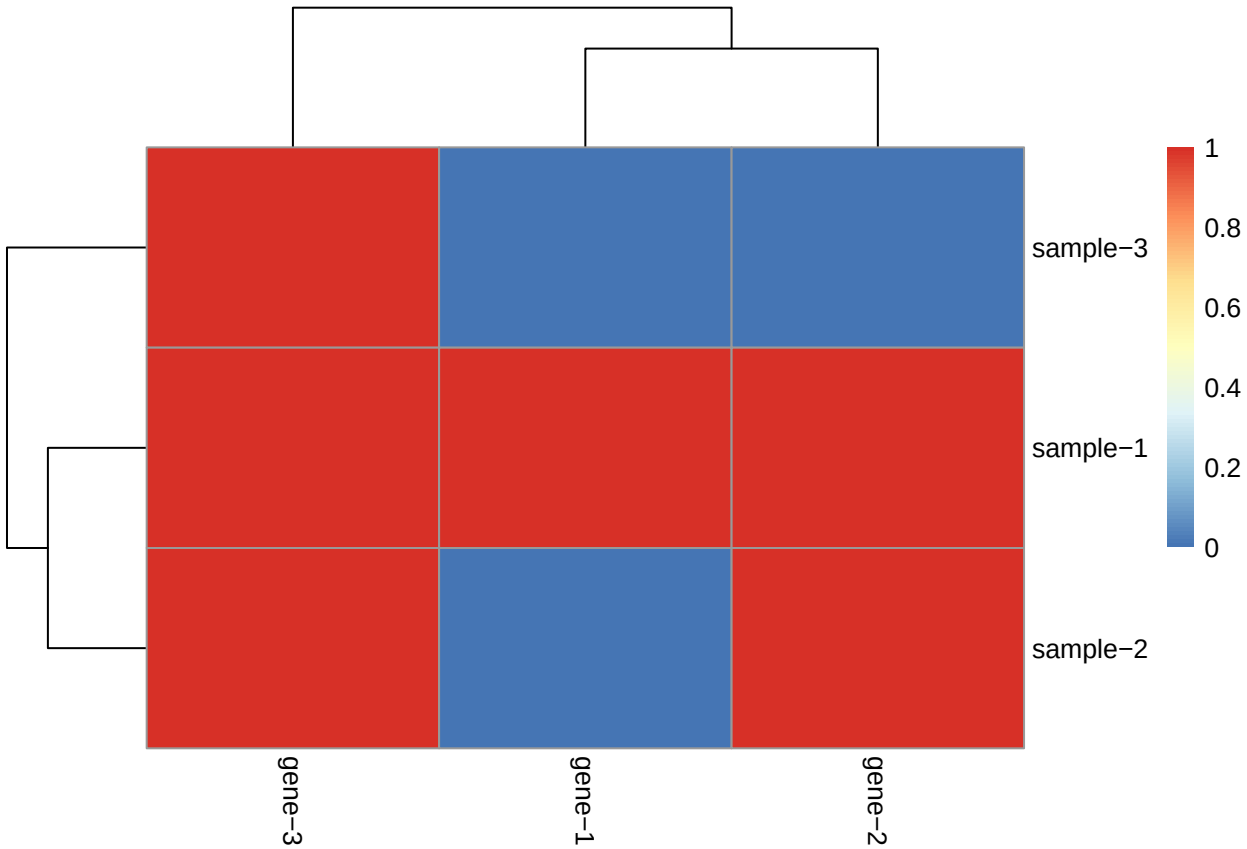
2020-05-08

```
# Libraries ---------------------------------------------------------------
library(tidyverse)
```

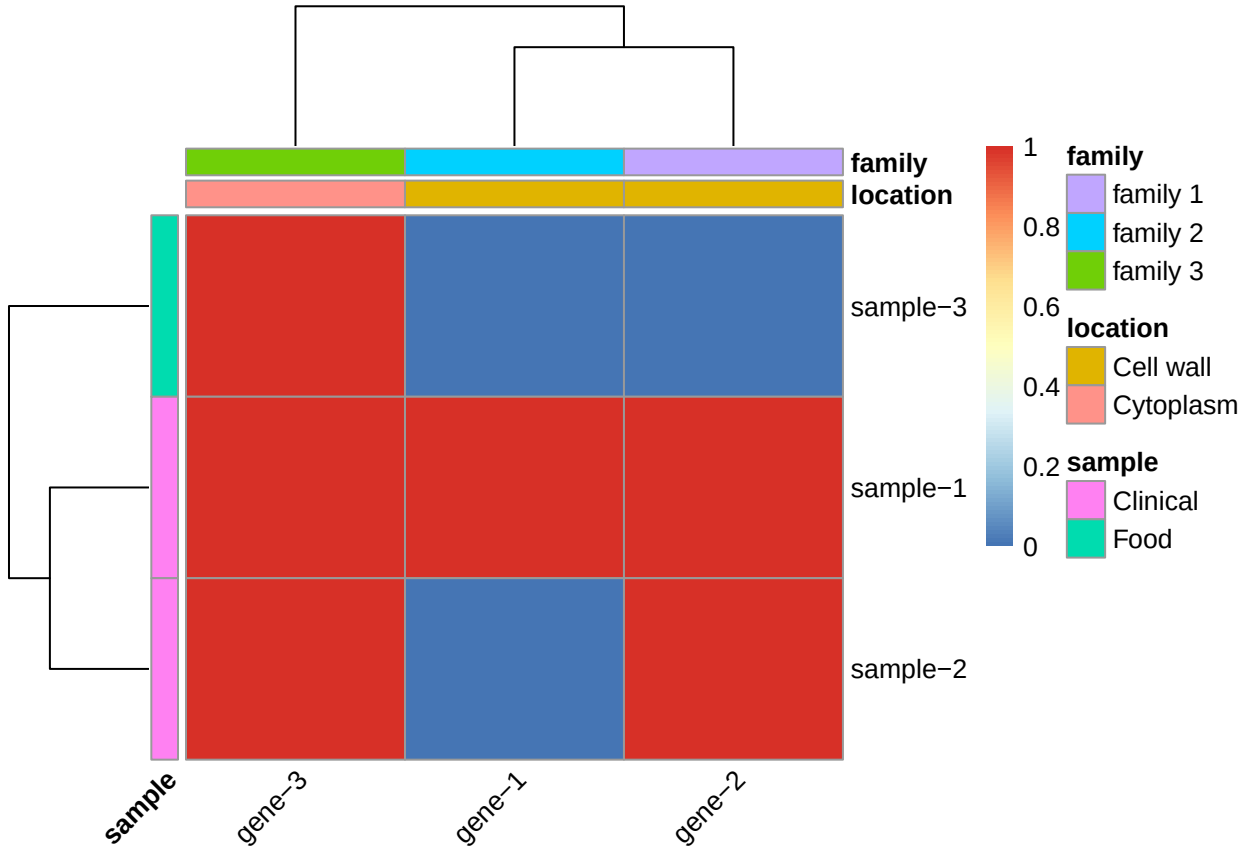
```
## -- Attaching packages ------------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(pheatmap)
library(RColorBrewer)

# Transform the data ------------------------------------------------------

example_data_binary <- matrix(data = c(1,0,0,1,1,0,1,1,1),
                              nrow = 3,
                              ncol = 3)
# set the names of columns and rows, if the matrix/dataframe has names in rows and columns avoid this.
colnames(example_data_binary) <- paste("gene", 1:3, sep = "-")
rownames(example_data_binary) <- paste("sample", 1:3, sep = "-")

# Add a characteristic to columns-----------------------------------------------------
# set the characteristic for the column
columns_characteristic <- data.frame("location" = c("Cell wall","Cell wall","Cytoplasm"),
                                     "family" = c("family 2", "family 1", "family 3"))
# To attached the characteristic to the genes you have to set the rownames equally to those in your ori
rownames(columns_characteristic) <- colnames(example_data_binary)

# Add a characteristic for rows -------------------------------------------

# set the characteristic for the rows, in this case the rownames of the data.frame must be the same to
rows_characteristic <- data.frame("sample type" = c("Clinical", "Clinical", "Food"))
#if you insert a name with space the data.frame will convert to "sample.type"
colnames(rows_characteristic)[1] <- "sample"
rownames(rows_characteristic) <- rownames(example_data_binary)
```

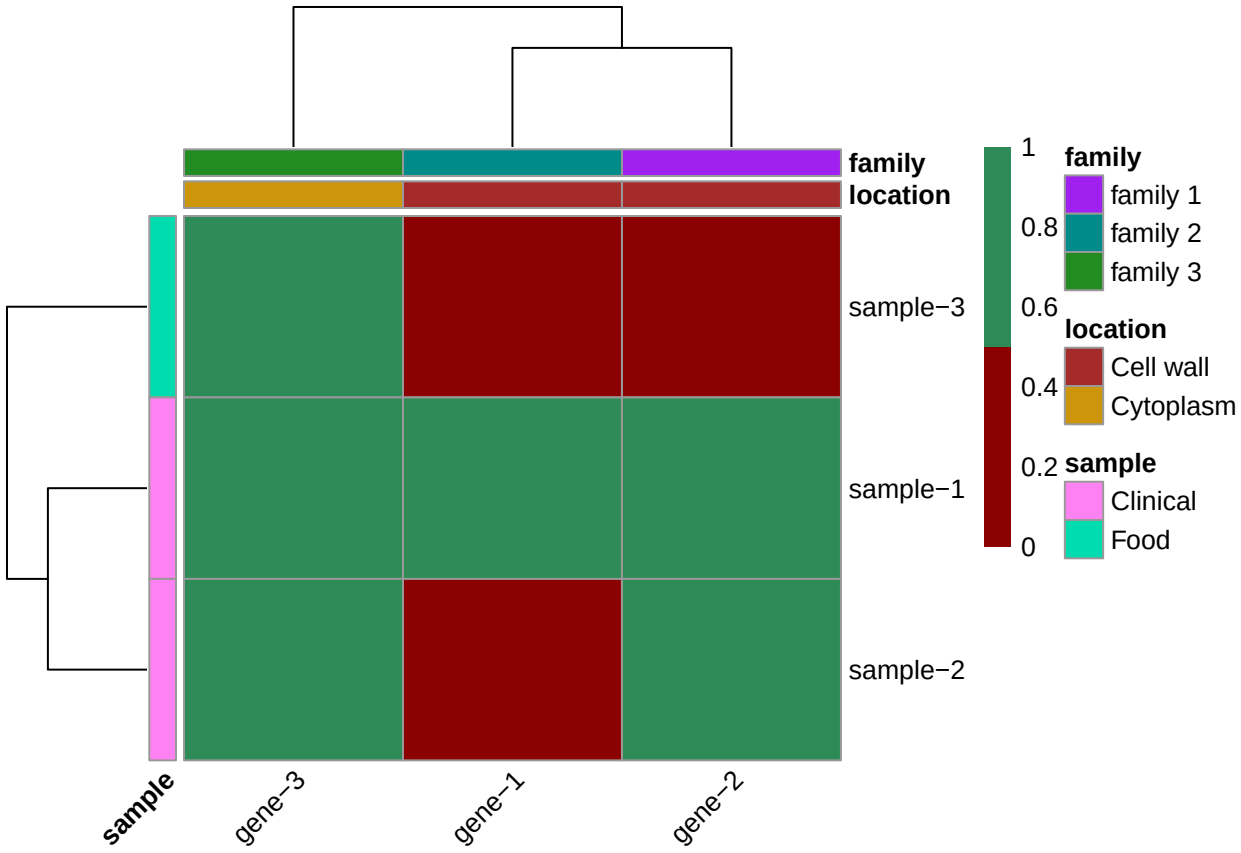
```r
# Set manually the colors for the characteristics created -----------------

# You can set colors to each of this parameters by two ways:
# In two steps
location_color <- c("brown","darkgoldenrod3")
names(location_color) <- unique(columns_characteristic$location)
# In one step
family_color <- setNames(object = c("purple","cyan4","forestgreen"),
                         nm = unique(columns_characteristic$family) %>% sort())
sample_color <- setNames(object = c("firebrick","lightgreen"),
                         nm = unique(rows_characteristic$`sample type`))

# Pheatmap needs to have this color palette in a list with the name of the tracks you want to visualize

pheatmap_colors <- list("location" = location_color,
                        "sample type" = sample_color,
                        "family" = family_color)

# Plot the heatmap --------------------------------------------------------

# Basic heatmap
pheatmap(mat = example_data_binary)
```



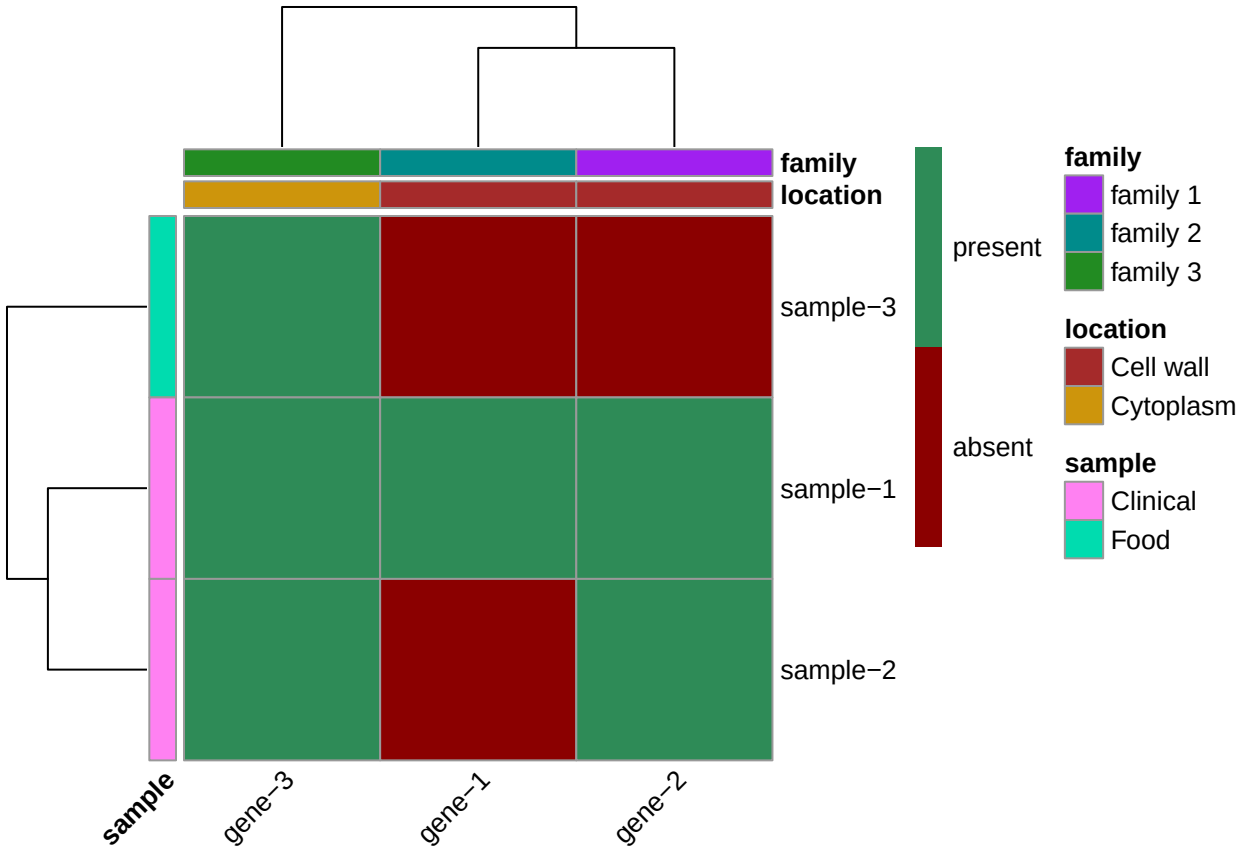
```
# Customize a bit the heatmap
pheatmap(mat = example_data_binary,
  angle_col = 45, # change the angle of the column labels,
  annotation_col = columns_characteristic, # pass the column characteristics data.frame
  annotation_row = rows_characteristic)
```



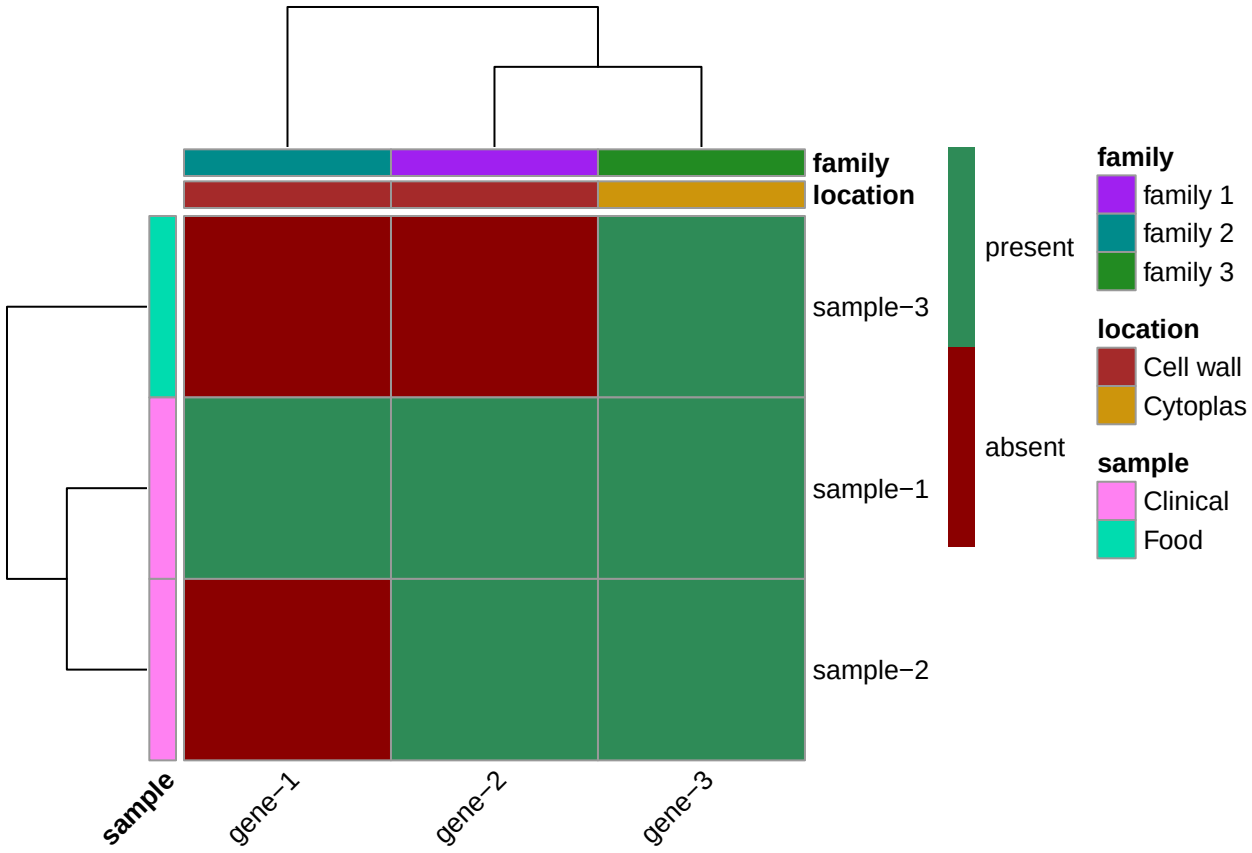

```
# We can do better if change the colors manually
pheatmap(mat = example_data_binary,
         angle_col = 45,
         annotation_col = columns_characteristic,
         annotation_row = rows_characteristic,
         annotation_colors = pheatmap_colors, # set the colors that you want for tracks,
         color = c("darkred","seagreen")) # change the heatmap cell color
```



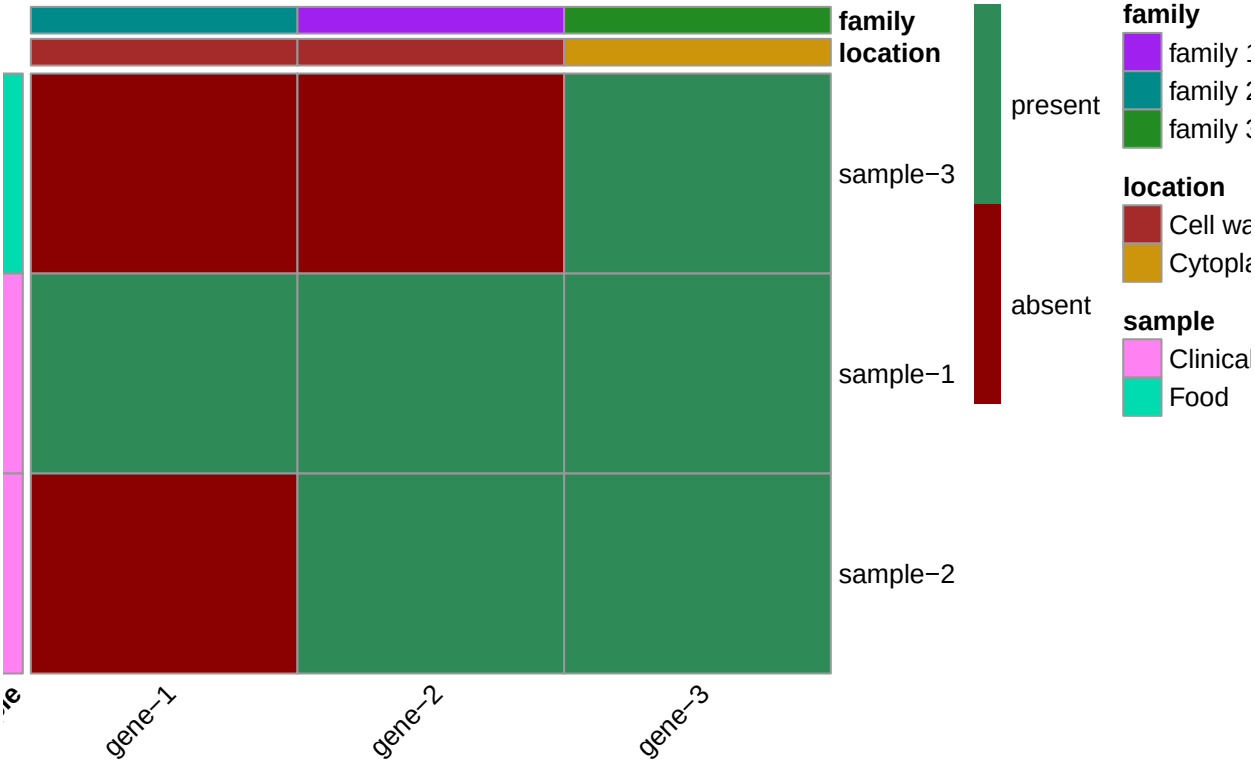

```
# Change the legend to be concordant with a "presence / absence" matrix

pheatmap(mat = example_data_binary,
         angle_col = 45,
         annotation_col = columns_characteristic,
         annotation_row = rows_characteristic,
         annotation_colors = pheatmap_colors,
         color = c("darkred","seagreen"),
         legend_breaks = c(0.25,0.75), # Set the legend to have two colors
         legend_labels = c("absent","present")) # labels to the legend
```




```
# Change the clustering method to group the columns and rows properly

pheatmap(mat = example_data_binary,
         angle_col = 45,
         annotation_col = columns_characteristic,
         annotation_row = rows_characteristic,
         annotation_colors = pheatmap_colors,
         color = c("darkred","seagreen"),
         legend_breaks = c(0.25,0.75),
         legend_labels = c("absent","present"),
         clustering_distance_rows = "binary", # binary distance is appropiate for 0 v.s 1
         clustering_distance_cols = "binary",
         clustering_method = "average") # linkage criteria can be minimum, maximum and average
```




```
# If you don't like have the columns or rows clustered just type
# cluster_rows = F / cluster_cols = F

# If you don't want to draw the trees in the margins you will need to set the cell width and height and

image <- pheatmap(mat = example_data_binary,
          angle_col = 45,
          annotation_col = columns_characteristic,
          annotation_row = rows_characteristic,
          annotation_colors = pheatmap_colors,
          color = c("darkred","seagreen"),
          legend_breaks = c(0.25,0.75),
          legend_labels = c("absent","present"),
          clustering_distance_rows = "binary",
          clustering_distance_cols = "binary",
          clustering_method = "average",
          treeheight_row = 0, # draw the tree of rows but with a height of 0
          treeheight_col = 0, # draw the tree of cols but with a height of 0
          cellwidth = 100, # width of the cell
          cellheight = 75) # height of the cell
```




```
dev.off()
```

```
## null device
##           1
```

```
# You can save the image, and again try some width/height numbers that satisfied you

#ggsave(filename = "/Path/where/you/want/to/save/name_of_file.pdf",
       # device = "pdf",
       # width = 10,
       # height = 8,
       # )
```