Decontaminer Visualization support

Code ▾

Amarinder Singh Thind

2022 Dec 07

- set working directory and load input file
- select sub set based on coulmn number or name
- preparation of data format for plots

set working directory and load input file

Hide

```
setwd("test_html/")
originaldata <- read.table("barPlotInfo_ge_CT_5_breast.txt",sep='\t', row.names=1, header=TRUE) ## should be data frame
```

select sub set based on coulmn number or name

Hide

```
 newdata_1=subset(originaldata, select=c(1:10,22,23))  ## col number based OR
 #newdata_1=subset(originaldata, select=c("Breast.03","Breast.08","Breast.100")) ##name based
```

Mean greater than threshold

Hide

```
newdata_1=newdata_1[rowSums(newdata_1) >0.1 ,drop=FALSE, ] #take only those rows which are greater than zero

allsample_threshhold <- 1 ## Define threshold for average count > threshold

data_wt=newdata_1[(rowSums(newdata_1)/nrow(newdata_1))>= allsample_threshhold,drop=FALSE, ]
nor <- data_wt
data_wt <- as.data.frame(t(data_wt))
```

preparation of data format for plots

Hide

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Hide

```
library('dplyr') ##HAS COUNT FUNCTION

data_wt$Sample <-  row.names(data_wt)
data_wt <- data_wt %>%
  pivot_longer(!Sample, names_to = "Species", values_to = "count") # Gathering the columns to have normalized counts to a single column
```

Lets Plot

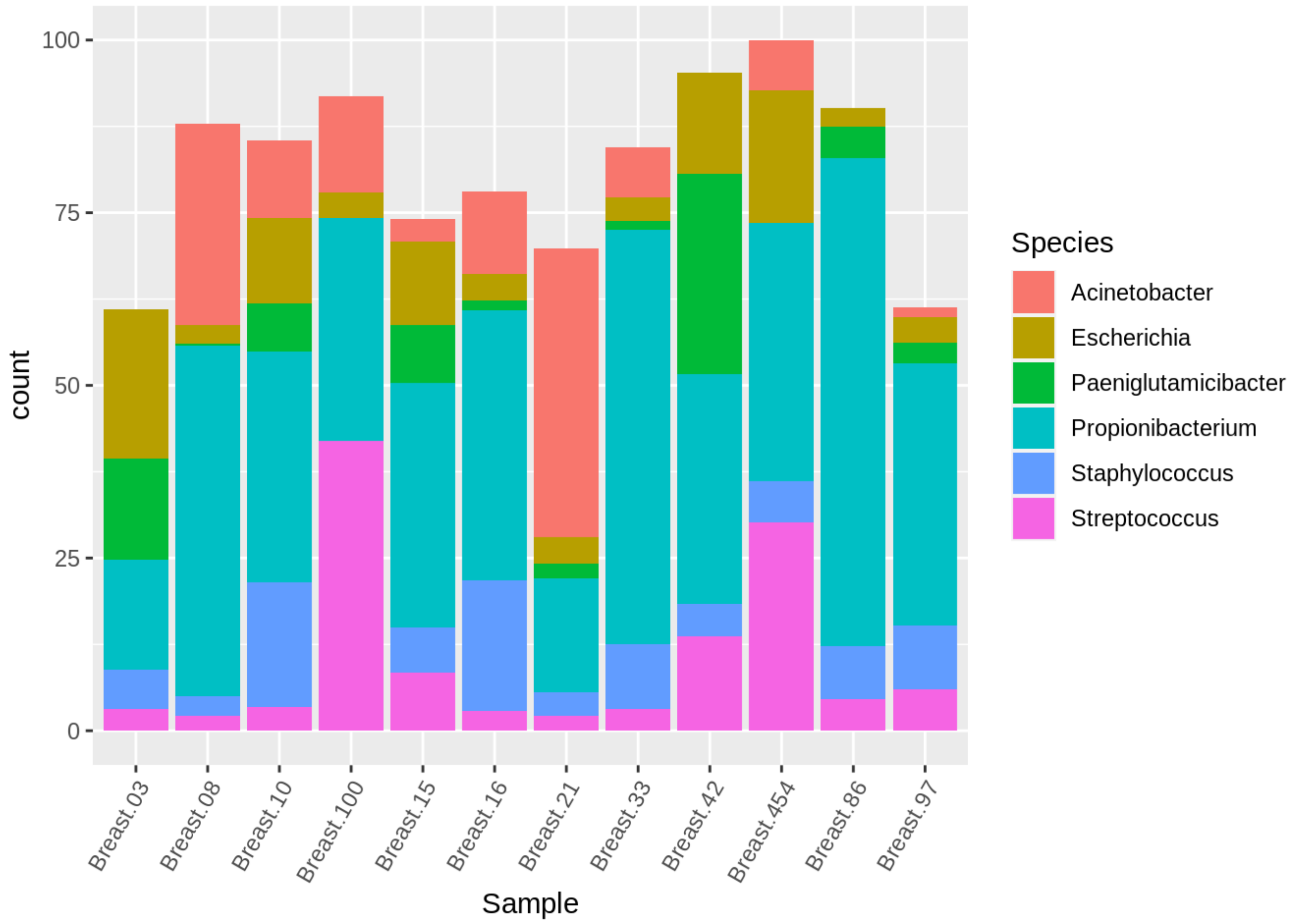
Hide

```
library(ggplot2)
library(RColorBrewer)

## setting 2 different color variables

my_palette <- colorRampPalette(c("#179493","#76c286", "#ebdc96", "#ec9173","#d0587e"), alpha=TRUE)(n=20)
heat_colors <- brewer.pal(6, "Reds") ##"Greens"  ##"Blues"
```

First : Stack plot with count

Hide

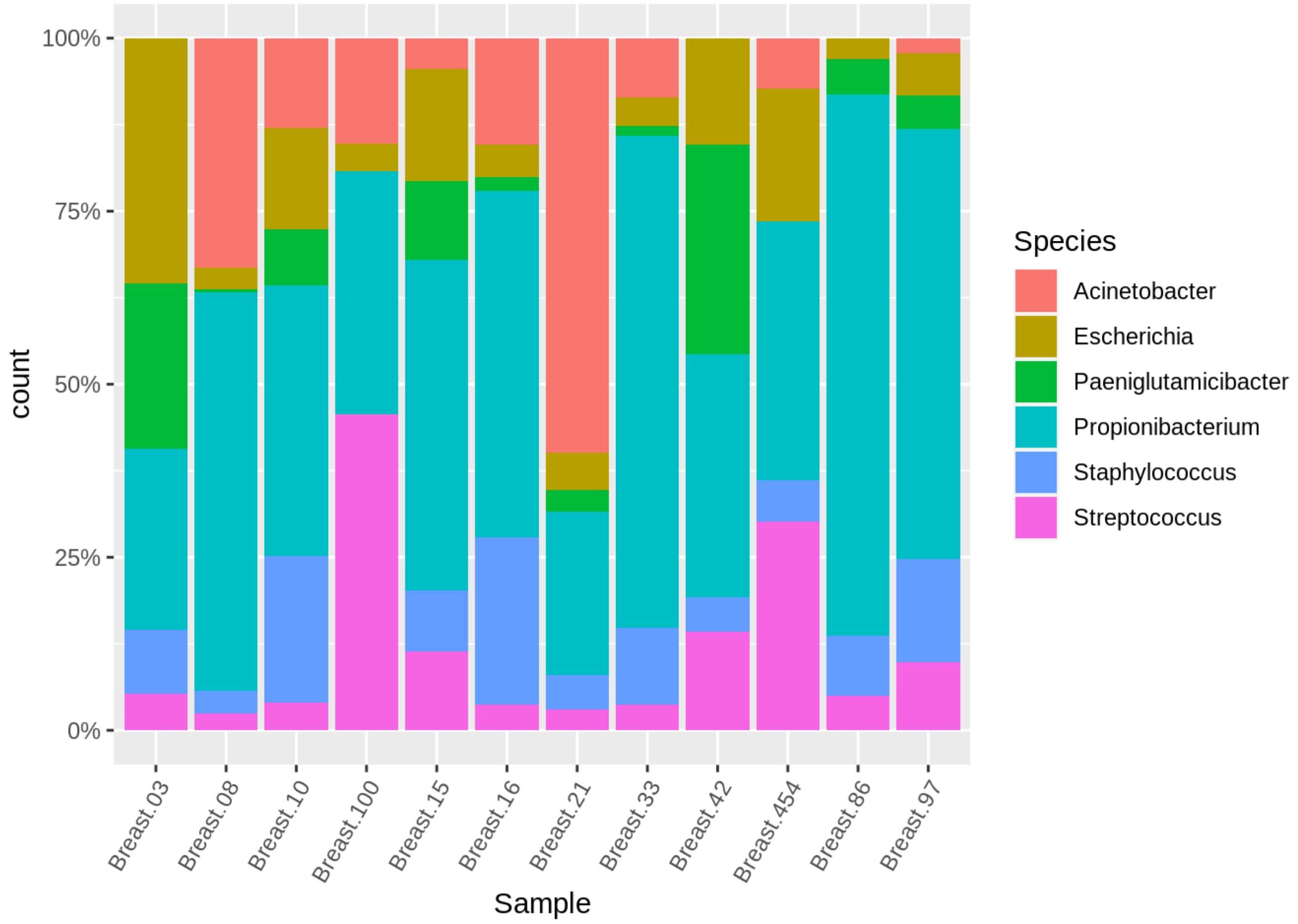
```
ggplot(data_wt, aes(x = Sample, y = count, fill = Species)) +
  geom_bar(stat = "identity")  + theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust=1))
```



Second : Stack plot with percent

Hide

```
ggplot(data_wt, aes(x = Sample, y = count, fill = Species)) +
  geom_col(position = "fill") +
  scale_y_continuous(labels = scales::percent)  + theme(axis.text.x = element_text(an
gle = 60, vjust = 1, hjust=1))
```

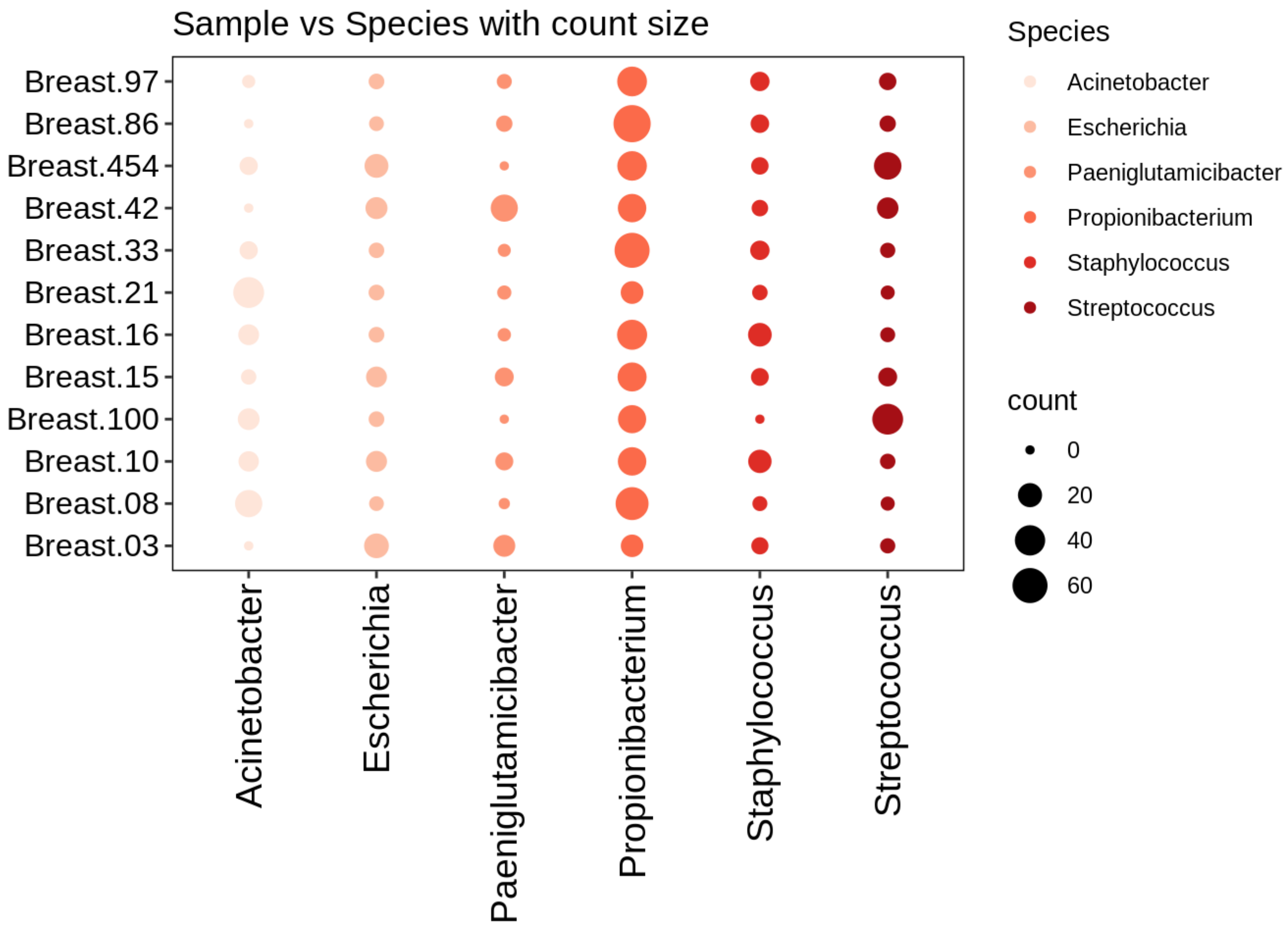


Third :Dot Plot (Sample vs Species with count size)

Hide

```
ggplot(data_wt,aes(x=Species,y=Sample )) +
  geom_point(aes(size=count,  color = Species))+
  scale_color_manual(values = heat_colors )+              ### in case heat_color does
nt work replace with manual or other like c("red","blue")
  #scale_color_gradientn('Log2 mean (Molecule 1, Molecule 2)', colors=my_palette) +
  ggtitle("Sample vs Species with count size ") +
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major = element_blank(),
        axis.text=element_text(size=14, colour = "black"),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        axis.text.y = element_text(size=12, colour = "black"),
        axis.title=element_blank(),
        panel.border = element_rect(size = 0.5, linetype = "solid", colour = "black"
))
```

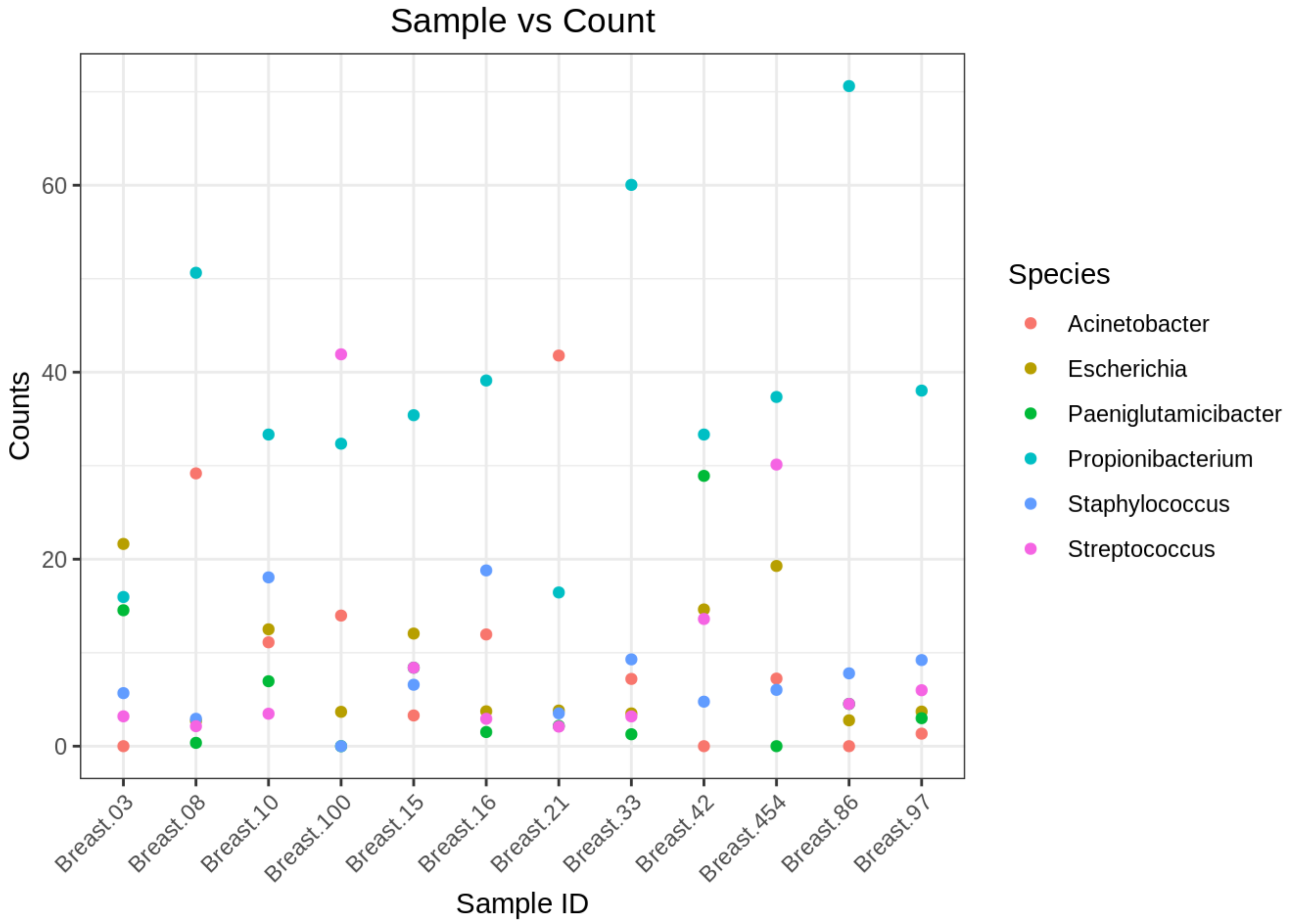
```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.
0.
## ℹ Please use the `linewidth` argument instead.
```



Fourth : Dot Plot (Sample vs Count)

Hide

```
ggplot(data_wt) +
  geom_point(aes(x = Sample , y = count, color = Species )) +
  ## scale_y_log10() +  ##want to scale it or not??
  xlab("Sample ID") +
  ylab("Counts") +
  ggtitle("Sample vs Count") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5))
```



Fifth : Heat Map ##############3

Hide

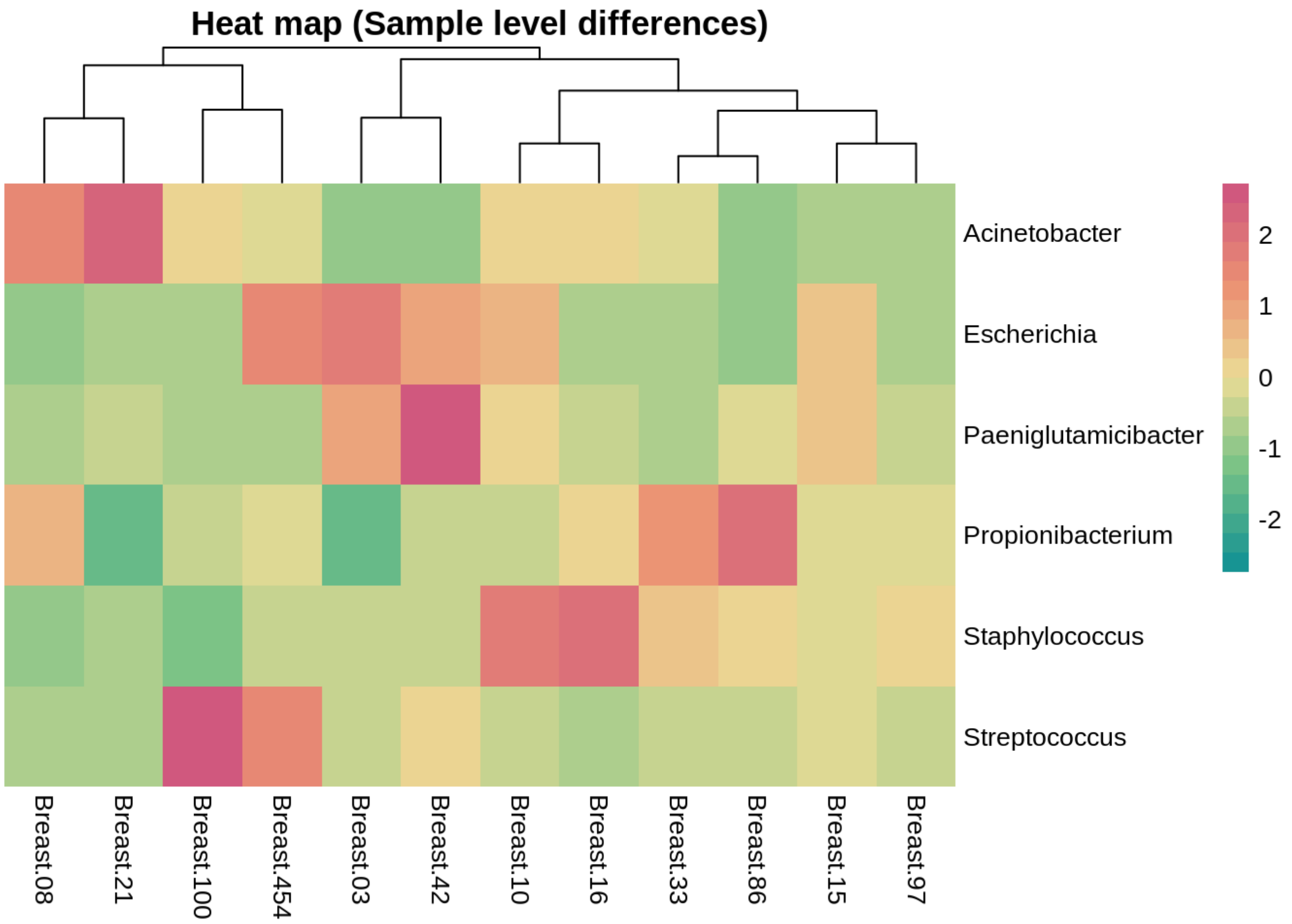
```
library(ComplexHeatmap)
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 2.12.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite:
## Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##   genomic data. Bioinformatics 2016.
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
```

Hide

```
### Run pheatmap
pheatmap::pheatmap( nor,
                    color = my_palette,
                    cluster_cols  = T,
                    cluster_rows = F,
                    show_rownames = T,
                    border_color = NA,
                    fontsize = 10,
                    scale = "row", #row #column #none  ## row wise differences clarit
y (i.e b/w samples)
                    fontsize_row = 10,
                    height = 20,
                    main= "Heat map (Sample level differences)"
)
```


Heat map (Sample level differences)
Acinetobacter
Escherichia
Paeniglutamicibacter
Propionibacterium
Staphylococcus
Streptococcus
Breast.08
Breast.21
Breast.100
Breast.454
Breast.03
Breast.42
Breast.10
Breast.16
Breast.33
Breast.86
Breast.15
Breast.97