

Microbial Biogeography of Restroom Surfaces

Thibaut Thalamas, Konstantinn Bonnet

12/15/2020

Setup

Load packages.

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(tidyr)
library(tibble)
library(ggplot2)
library(ggdendro)
library(RColorBrewer)
```

Set a general *ggplot2* theme to use throughout.

```
th <- theme_classic() +
  theme(panel.grid.major=element_line(size=0.5, linetype="solid", colour="#e6e6e6"),
        panel.grid.minor=element_line(size=0.25, linetype="solid", colour="#e6e6e6"))
theme_set(th)
```

Read data and prepare data

Read the OTU table and cast counts to numeric, since `cor()` and others require numeric data.

```
# read otu table and cast counts to numeric, cor() and others require numeric data
otu <- read_delim("1335.vdb.tab", header=TRUE) %>%
  mutate_if(is.integer, as.numeric)
str(otu)
```

```
## 'data.frame':   4105 obs. of  16 variables:
## $ OTUNumber      : chr  "0tu00001" "0tu00002" "0tu00003" "0tu00004" ...
```

```
## $ OTUConTaxonomy      : chr "Bacteria(100);Actinobacteria(100);Actinobacteria(100);Actinomycetales(100)"
## $ door_in_1           : num 1381 20 8 6 0 ...
## $ door_in_2           : num 1032 17 47 27 29 ...
## $ faucet_handle_1     : num 335 1686 6 5 0 ...
## $ faucet_handle_2     : num 577 3 418 6 648 160 11 31 0 116 ...
## $ sink_floor_1        : num 38 411 15 452 2 8 119 0 8 5 ...
## $ sink_floor_2        : num 36 564 8 275 8 8 243 7 5 5 ...
## $ soap_dispenser_1   : num 1115 128 84 63 3 ...
## $ stall_in_1          : num 598 9 45 0 32 115 10 225 0 0 ...
## $ toilet_floor_1      : num 45 430 6 236 1 24 112 4 14 11 ...
## $ toilet_floor_2      : num 87 707 30 251 5 7 234 3 13 11 ...
## $ toilet_flush_handle_1: num 13 0 3 9 0 3 4 14 0 2 ...
## $ toilet_flush_handle_2: num 329 25 283 51 206 130 10 49 17 61 ...
## $ toilet_seat_1       : num 76 258 41 5 7 32 29 56 32 54 ...
## $ toilet_seat_2       : num 156 0 1059 3 95 ...
```

Preliminary analysis and data manipulation

First, get the names of phyla and taxa by splitting the *OTUConTaxonomy* column, containing the assigned taxonomy for each OTU.

```
# for each row of the otu table
tax <- apply(otu, 1, function(x)
  # select OTUConTaxonomy column
  x[2] %>%
    # split strings and get the result, returned inside a list
    strsplit(";", " ") %>%
    # subset the list
    "[1]" %>%
    # remove assignment score from the name
    sub(pattern="\\(.*", replacement="")
)
# transpose so OTUs are again the rows
tax <- t(tax)
str(tax)
```

```
## chr [1:4105, 1:6] "Bacteria" "Bacteria" "Bacteria" "Bacteria" "Bacteria" ...
```

```
# bind the new phylum and taxon column to the original table
otu <- cbind(otu, data.frame(Genus=tax[,2], Taxon=tax[,6]))
str(otu)
```

```
## 'data.frame': 4105 obs. of 18 variables:
## $ OTUNumber           : chr "Otu00001" "Otu00002" "Otu00003" "Otu00004" ...
## $ OTUConTaxonomy      : chr "Bacteria(100);Actinobacteria(100);Actinobacteria(100);Actinomycetales(100)"
## $ door_in_1           : num 1381 20 8 6 0 ...
## $ door_in_2           : num 1032 17 47 27 29 ...
## $ faucet_handle_1     : num 335 1686 6 5 0 ...
## $ faucet_handle_2     : num 577 3 418 6 648 160 11 31 0 116 ...
## $ sink_floor_1        : num 38 411 15 452 2 8 119 0 8 5 ...
## $ sink_floor_2        : num 36 564 8 275 8 8 243 7 5 5 ...
## $ soap_dispenser_1   : num 1115 128 84 63 3 ...
## $ stall_in_1          : num 598 9 45 0 32 115 10 225 0 0 ...
## $ toilet_floor_1      : num 45 430 6 236 1 24 112 4 14 11 ...
## $ toilet_floor_2      : num 87 707 30 251 5 7 234 3 13 11 ...
## $ toilet_flush_handle_1: num 13 0 3 9 0 3 4 14 0 2 ...
```

```
## $ toilet_flush_handle_2: num 329 25 283 51 206 130 10 49 17 61 ...
## $ toilet_seat_1 : num 76 258 41 5 7 32 29 56 32 54 ...
## $ toilet_seat_2 : num 156 0 1059 3 95 ...
## $ Genus : chr "Actinobacteria" "Proteobacteria" "Actinobacteria" "Actinobacteria" .
## $ Taxon : chr "Propionibacterium" "Enhydrobacter" "Corynebacterium" "Arthrobacter"
```

For analysis further on, extract the only the counts from the table, and convert to a matrix: handy for some of the computations.

```
otumat <- otu %>%
  # the counts are the only numeric columns
  select_if(is.numeric) %>%
  as.matrix
str(otumat)
```

```
## num [1:4105, 1:14] 1381 20 8 6 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:14] "door_in_1" "door_in_2" "faucet_handle_1" "faucet_handle_2" ...
```

Next, we'll need the summed counts by taxon across OTUs, and get the top 19 taxons in terms of abundance. The very first element is actually *unclassified*, the unassigned OTUs, and we'll discard it.

```
# this same construct will be used later:
# take the numeric count matrix
taxtotals <- otumat %>%
  # for each of its columns (sites)
  apply(2, function(x)
    # for each unique taxon (rows)
    sapply(unique(otu$Taxon), function(t)
      # sum its abundance
      sum(x[otu$Taxon == t])
    )
  # this returns total abundance per taxon per site
  ) %>%
  # sum abundance across sites => table with total abundance per taxon
  apply(1, sum)
str(taxtotals)
```

```
## Named num [1:357] 6003 4315 4941 1458 1589 ...
## - attr(*, "names")= chr [1:357] "Propionibacterium" "Enhydrobacter" "Corynebacterium" "Arthrobacter"
```

```
# order taxons by descending abundance
taxtotals <- taxtotals[order(-taxtotals)]
# then extract the top 19 taxons (removing unclassified)
toptax <- taxtotals[2:20]
toptax
```

| | | | |
|----------------------|------------------|---------------|----------------|
| ## Propionibacterium | Corynebacterium | Enhydrobacter | Staphylococcus |
| ## 6003 | 4941 | 4315 | 1589 |
| ## Arthrobacter | Sphingomonas | Acinetobacter | Pseudomonas |
| ## 1458 | 1242 | 1140 | 805 |
| ## Aquabacterium | Anaerococcus | Roseomonas | Streptococcus |
| ## 787 | 674 | 536 | 535 |
| ## Finegoldia | Chryseobacterium | Kocuria | Hymenobacter |
| ## 434 | 382 | 374 | 345 |
| ## Friedmanniella | Methylobacterium | Sphingobium | |

Phylum composition by site

We'll first explore phylum composition by site. Using the previous manipulations, we'll first compute a slightly prettier color palette for each phylum, then output a stacked percent barchart, showing relative proportion of each phylum in each site.

Color palette

Here, we use ColorBrewer palettes for good constructs and for categorical data, since those sets have the most colors. We have more phylum (18) than colors in the set (12), so we'll interpolate some additional colors.

```
# number of unique phylum, the number of colors needed
ngenus <- length(unique(otu$Genus))
# extract the colors of the Set1 brewer color palette
# this gives a harmless warning since we request too many colors
pal.g <- brewer.pal(ngenus, "Set1")

## Warning in brewer.pal(ngenus, "Set1"): n too large, allowed maximum for palette Set1 is 9
## Returning the palette you asked for with that many colors

# interpolate for the remaining elements
# colorRampPalette returns a function which we use directly to extract our colors
cmap.g <- colorRampPalette(pal.g, bias=1, interpolate="spline")(ngenus)
cmap.g

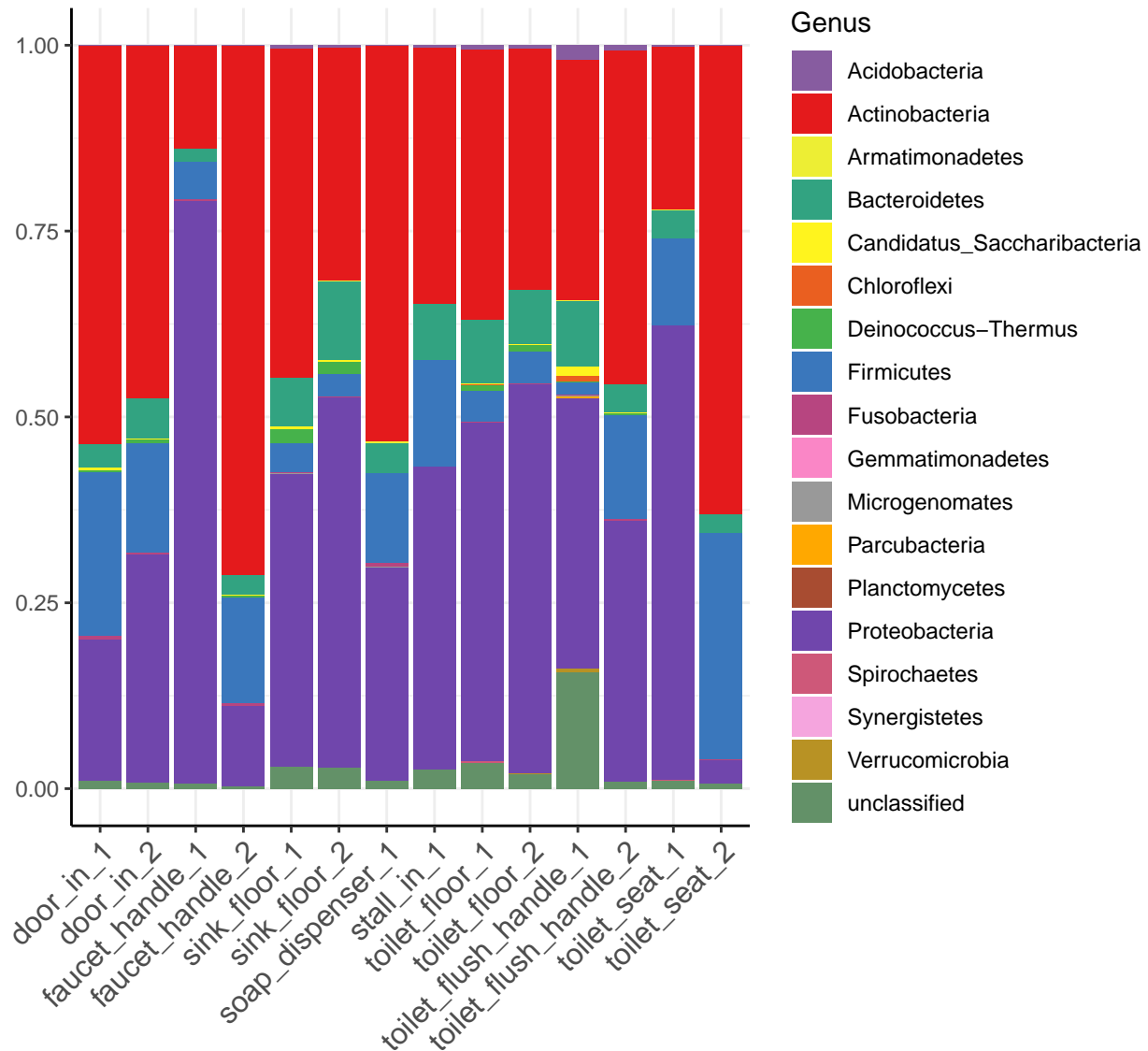
## [1] "#E41A1C" "#7046AC" "#3A77BC" "#32A381" "#46B24B" "#639168" "#875CA0"
## [8] "#B74580" "#EA5F20" "#FFA800" "#FFF41E" "#EDEE34" "#B89224" "#A84C31"
## [15] "#CE5879" "#FA86C6" "#F5A5DE" "#999999"
```

Plot

We have to transform our OTU table into long format for use with *ggplot2*. Since we don't reuse this result, we'll just pipe it directly to *ggplot()*.

Finally, we save the *ggplot* object to a png file. Rendering to pdf had different results on various machines, as could possibly be seen in this document.

```
# stacked percent barchart
bar.g <- otu %>%
  # get count columns and pivot to long format (similar to melt in reshape2)
  # tidyverse matches() is used to select the columns via a regex
  # site column will contains the sample site, and abundance the abundance values
  pivot_longer(cols=matches("^a-z"), ignore.case=FALSE, names_to="site", values_to="abundance") %>%
  # x=site, y=abundance and use phylum as group colors for the bar's fill
  ggplot(aes(site, abundance, fill=Genus)) +
  # output percent stacked barchart
  geom_bar(position="fill", stat="identity") +
  # rotate xlab and remove x title
  theme(axis.text.x=element_text(angle=45, vjust=1, size=12, hjust=1),
        axis.title=element_blank()) +
  # manually set the colors for each phylum from the palette
  scale_fill_manual(values=cmap.g[order(unique(otu$Genus))])
bar.g
```



```
# save it
ggsave("plots/stacked.genus.by.site.png", bar.g, width=12, height=10)
```

Taxonomic composition by site

Next, we'll do the same, but this time for taxa. These are more precise and familiar and can provide interesting results.

To make a prettier and more informative plot, there are several complicated steps involved.

To easily view which phylum the taxa belong to, we'd like to group them visually. An intuitive way to do this is to make gradients of one color for each phylum, since there are no more than a few of them, and the taxa are few enough that there still will be enough contrast to distinguish each of them.

To this end, we need to create the color palette manually for each, but in a scriptable and extensible way.

Creating the color palettes

We'll have to build a correspondence table with the top 19 taxa found earlier.

First, we extract the phylum and taxon columns from the taxonomy table computed earlier.

```
# filter the taxonomy table by the names of the top taxa, get phylum (2) and taxon (6) columns
corr <- tax[tax[,6] %in% names(toptax),][,c(2,6)]
# only take unique rows, then convert to data frame and set appropriate column names
corr <- unique(corr) %>%
  as.data.frame %>%
  # "family" seems more appropriate than "genus" here
  rename(Family=V1, Taxon=V2)
corr
```

| ## | Family | Taxon |
|-------|----------------|-------------------|
| ## 1 | Actinobacteria | Propionibacterium |
| ## 2 | Proteobacteria | Enhydrobacter |
| ## 3 | Actinobacteria | Corynebacterium |
| ## 4 | Actinobacteria | Arthrobacter |
| ## 5 | Firmicutes | Staphylococcus |
| ## 6 | Proteobacteria | Acinetobacter |
| ## 7 | Proteobacteria | Aquabacterium |
| ## 8 | Proteobacteria | Sphingomonas |
| ## 9 | Firmicutes | Finegoldia |
| ## 10 | Proteobacteria | Pseudomonas |
| ## 11 | Bacteroidetes | Chryseobacterium |
| ## 12 | Proteobacteria | Roseomonas |
| ## 13 | Actinobacteria | Kocuria |
| ## 14 | Firmicutes | Streptococcus |
| ## 15 | Actinobacteria | Friedmanniella |
| ## 16 | Firmicutes | Anaerococcus |
| ## 17 | Proteobacteria | Sphingobium |
| ## 18 | Proteobacteria | Methylobacterium |
| ## 19 | Bacteroidetes | Hymenobacter |

Next, we use ordered factors to force ordering by phylum, then taxon. This is necessary to preserve this order later since we further manipulate the tables, and have to make sure colors are correctly attributed.

```
# take our correspondence table
corr <- corr %>%
  # convert the family column to an ordered factor according to abundance and importance
  # the levels are from what was obtained above
  mutate(Family=factor(Family,
    levels=c("Proteobacteria", "Actinobacteria", "Firmicutes", "Bacteroidetes"),
    ordered=TRUE)) %>%
  # sort by family then taxon
  arrange(Family, Taxon) %>%
  # now freeze the new taxon order
  mutate(Taxon=factor(Taxon, levels=unique(Taxon), ordered=TRUE))
corr
```

| ## | Family | Taxon |
|------|----------------|------------------|
| ## 1 | Proteobacteria | Acinetobacter |
| ## 2 | Proteobacteria | Aquabacterium |
| ## 3 | Proteobacteria | Enhydrobacter |
| ## 4 | Proteobacteria | Methylobacterium |
| ## 5 | Proteobacteria | Pseudomonas |
| ## 6 | Proteobacteria | Roseomonas |
| ## 7 | Proteobacteria | Sphingobium |

```
## 8 Proteobacteria Sphingomonas
## 9 Actinobacteria Arthrobacter
## 10 Actinobacteria Corynebacterium
## 11 Actinobacteria Friedmanniella
## 12 Actinobacteria Kocuria
## 13 Actinobacteria Propionibacterium
## 14 Firmicutes Anaerococcus
## 15 Firmicutes Finegoldia
## 16 Firmicutes Staphylococcus
## 17 Firmicutes Streptococcus
## 18 Bacteroidetes Chryseobacterium
## 19 Bacteroidetes Hymenobacter
```

Finally, we can attribute color gradients for each phylum, making sure each gets a different base color.

```
# add a new column with the colors for the ordering
corr$col <- c(
  # blue, red, green and violet/pink gradients
  # like before, we directly call the function returned with the appropriate number of colors we need
  colorRampPalette(c("lightblue", "darkblue"))(8),
  colorRampPalette(c("#ffff66", "red", "darkred"))(5),
  colorRampPalette(c("lightgreen", "darkgreen"))(4),
  colorRampPalette(c("violet", "blue"))(8)[1:2]
)
corr
```

```
##          Family          Taxon      col
## 1 Proteobacteria Acinetobacter #ADD8E6
## 2 Proteobacteria Aquabacterium #94B9D9
## 3 Proteobacteria Enhydrobacter #7B9ACC
## 4 Proteobacteria Methylobacterium #627BBF
## 5 Proteobacteria Pseudomonas #4A5CB2
## 6 Proteobacteria Roseomonas #313DA4
## 7 Proteobacteria Sphingobium #181E98
## 8 Proteobacteria Sphingomonas #00008B
## 9 Actinobacteria Arthrobacter #FFFF66
## 10 Actinobacteria Corynebacterium #FF7F33
## 11 Actinobacteria Friedmanniella #FF0000
## 12 Actinobacteria Kocuria #C40000
## 13 Actinobacteria Propionibacterium #8B0000
## 14 Firmicutes Anaerococcus #90EE90
## 15 Firmicutes Finegoldia #60C060
## 16 Firmicutes Staphylococcus #309230
## 17 Firmicutes Streptococcus #006400
## 18 Bacteroidetes Chryseobacterium #EE82EE
## 19 Bacteroidetes Hymenobacter #CC6FF0
```

Stacked percent barchart

We can now output the final plot using the colors we computed, the same way as before.

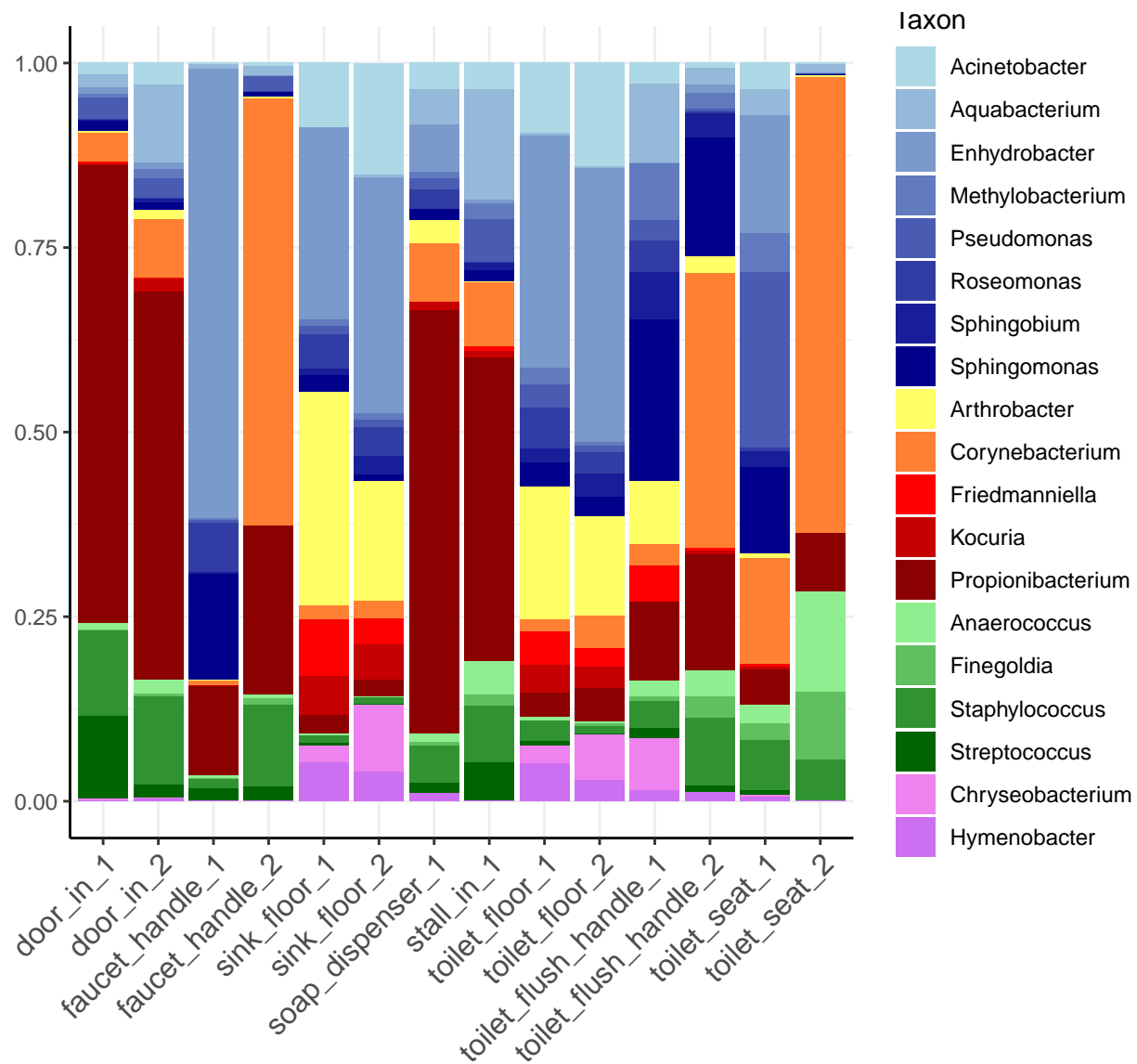
```
bar.t <- otu %>%
  # remove columns we don't care about
  select(-Genus, -starts_with("OTU")) %>%
  # extract rows for only top taxa
  filter(Taxon %in% names(toptax)) %>%
```

```

# convert to long format for ggplot as before
pivot_longer(cols=matches("[a-t]"), ignore.case=FALSE), names_to="site", values_to="abundance") %>%
# convert to ordered factor as with the correspondence table, using its order
# counterintuitive and timeconsuming to figure out, but very important to avoid any discrepancies..
mutate(Taxon=factor(Taxon, levels=levels(corr$Taxon), ordered=TRUE)) %>%
ggplot(aes(site, abundance, fill=Taxon)) +
  # stacked percent barchart
  geom_bar(position="fill", stat="identity") +
  # set our colors
  scale_fill_manual(values=corr$col) +
  # rotate x legend, remove x axis label
  theme(axis.text.x=element_text(angle=45, vjust=1, size=12, hjust=1),
        axis.title=element_blank())

```

bar.t



```

# save the plot
ggsave("plots/stacked.taxon.by.site.png", bar.t, width=12, height=10)

```


Phylum correlation map

It is of interest to evaluate links between phylum in our data. This can be done by computing the correlation between phylum, then visualizing it as a sort of heatmap.

Preparing the data

We first need to extract the totals for each phylum by site, since that's what we'll be calculating the correlation matrix on.

```
# use the abundance matrix computed at the beginning, cor() needs numerics
gentotals <- otumat %>%
  # for each site (column)
  apply(2, function(x)
    # for each OTU (row),
    # sum abundances of OTUs assigned to each individual phylum
    sapply(unique(otu$Genus), function(g)
      sum(x[otu$Genus == g])
    )
  )
head(gentotals)
```

```
##          door_in_1 door_in_2 faucet_handle_1 faucet_handle_2
## Actinobacteria    1669    1483           434           2224
## Proteobacteria     593     960          2445           340
## Firmicutes        686     457           155           440
## Bacteroidetes     100     170            54            84
## Deinococcus-Thermus  10      20             0            12
## unclassified       30      22            19             6
##          sink_floor_1 sink_floor_2 soap_dispenser_1 stall_in_1
## Actinobacteria    1382      978           1665          1077
## Proteobacteria    1229      1555            897          1272
## Firmicutes        123       94            378           446
## Bacteroidetes     201      333            127           235
## Deinococcus-Thermus  59       50             0             0
## unclassified       89       85            30            78
##          toilet_floor_1 toilet_floor_2 toilet_flush_handle_1
## Actinobacteria    1133      1013           1006
## Proteobacteria    1419      1634           1133
## Firmicutes        130       131            51
## Bacteroidetes     267      228            276
## Deinococcus-Thermus  23       28             5
## unclassified      106       60            486
##          toilet_flush_handle_2 toilet_seat_1 toilet_seat_2
## Actinobacteria    1400      685           1968
## Proteobacteria    1096      1906            100
## Firmicutes        433       363            946
## Bacteroidetes     120       117             80
## Deinococcus-Thermus  10        2             1
## unclassified       26       32            19
```

There is a high number of OTUs with single counts across sites, and many unassigned ones. Similarly, there are phyla with only a few corresponding rows, and very low counts. We could still compute a correlation with the rest, but it would be difficult to interpret. We set the threshold as low as possible. Two phyla are removed.

```
# calculate a frequency table
```

```
genfreq <- table(otu$Genus)
```

```
genfreq
```

```
##
##          Acidobacteria          Actinobacteria
##                79                1337
##          Armatimonadetes          Bacteroidetes
##                6                509
## Candidatus_Saccharibacteria          Chloroflexi
##                44                14
##          Deinococcus-Thermus          Firmicutes
##                35                381
##          Fusobacteria          Gemmatimonadetes
##                13                7
##          Microgenomates          Parcubacteria
##                2                6
##          Planctomycetes          Proteobacteria
##                5                1113
##          Spirochaetes          Synergistetes
##                3                1
##          Verrucomicrobia          unclassified
##                14                536
```

```
# filter the most infrequent phylum
```

```
gentotals <- gentotals[!rownames(gentotals) %in% names(genfreq)[genfreq < 5],]
```

```
head(gentotals)
```

```
##          door_in_1 door_in_2 faucet_handle_1 faucet_handle_2
## Actinobacteria    1669    1483          434          2224
## Proteobacteria     593     960         2445          340
## Firmicutes        686     457          155          440
## Bacteroidetes      100     170           54           84
## Deinococcus-Thermus  10      20            0           12
## unclassified       30      22           19            6
##          sink_floor_1 sink_floor_2 soap_dispenser_1 stall_in_1
## Actinobacteria    1382     978         1665         1077
## Proteobacteria    1229    1555          897         1272
## Firmicutes        123      94          378          446
## Bacteroidetes      201     333          127          235
## Deinococcus-Thermus  59      50            0            0
## unclassified       89      85           30           78
##          toilet_floor_1 toilet_floor_2 toilet_flush_handle_1
## Actinobacteria    1133     1013         1006
## Proteobacteria    1419     1634         1133
## Firmicutes        130      131           51
## Bacteroidetes      267     228          276
## Deinococcus-Thermus  23      28            5
## unclassified      106      60          486
##          toilet_flush_handle_2 toilet_seat_1 toilet_seat_2
## Actinobacteria    1400     685         1968
## Proteobacteria    1096     1906          100
## Firmicutes        433      363          946
## Bacteroidetes      120     117           80
## Deinococcus-Thermus  10        2            1
```

Computing correlation and ordering

We'll first compute an ordinary correlation matrix between phylum, which are the rows of the total abundance matrix.

```
gencor <- cor(t(gentotals))
```

Next, we'd like to group highly correlated phylum together. An easy way is to perform hierarchical clustering and use the clustering's order. We even get a dendrogram for easier visualization that way.

For this kind of data, Pearson distance, defined as $1 - \text{Pearson correlation}$ is appropriate, and Ward linkage method works very well. As indicated in the manual, *ward.D2* must be used, since our distances are not squared (or positive integers).

```
# perform the clustering
hcl <- hclust(as.dist(1 - gencor), method="ward.D2")
# extract order
horder <- rownames(gentotals)[hcl$order]
horder
```

```
## [1] "Fusobacteria"          "Actinobacteria"
## [3] "Firmicutes"           "Acidobacteria"
## [5] "Candidatus_Saccharibacteria" "unclassified"
## [7] "Chloroflexi"          "Parcubacteria"
## [9] "Verrucomicrobia"      "Armatimonadetes"
## [11] "Planctomycetes"       "Proteobacteria"
## [13] "Bacteroidetes"        "Deinococcus-Thermus"
## [15] "Gemmatimonadetes"
```

```
# convert to data appropriate for ggplot2, which gives us segment coordinates for drawing it directly
dhcl <- dendro_data(hcl)
# reorder the matrix
gencor <- gencor[horder,horder]
gencor
```

```
##               Fusobacteria Actinobacteria Firmicutes
## Fusobacteria      1.0000000      0.52231230  0.4185781
## Actinobacteria     0.5223123      1.00000000  0.6388938
## Firmicutes         0.4185781      0.63889378  1.0000000
## Acidobacteria     -0.4434068     -0.30588683 -0.5067274
## Candidatus_Saccharibacteria -0.2770136     -0.14812119 -0.4046593
## unclassified       -0.4212899     -0.25654800 -0.4637358
## Chloroflexi        -0.2780714     -0.18032203 -0.3610519
## Parcubacteria      -0.2832168     -0.18421438 -0.3532849
## Verrucomicrobia    -0.3035044     -0.19970302 -0.3837630
## Armatimonadetes     0.1685259     -0.04009420 -0.1773478
## Planctomycetes     0.1167320      0.04730139 -0.3280876
## Proteobacteria     -0.3880156     -0.95048641 -0.7020287
## Bacteroidetes      -0.6488839     -0.33141043 -0.6172501
## Deinococcus-Thermus -0.3478978     -0.03705856 -0.4772676
## Gemmatimonadetes   -0.4600615     -0.29742251 -0.6147621
##               Acidobacteria Candidatus_Saccharibacteria
## Fusobacteria      -0.44340677      -0.27701357
## Actinobacteria     -0.30588683      -0.14812119
## Firmicutes         -0.50672741      -0.40465930
```

| | | | |
|--------------------------------|-----------------|-----------------|---------------------|
| ## Acidobacteria | 1.00000000 | | 0.89946659 |
| ## Candidatus_Saccharibacteria | 0.89946659 | | 1.00000000 |
| ## unclassified | 0.93818310 | | 0.94984972 |
| ## Chloroflexi | 0.91273061 | | 0.92724326 |
| ## Parcubacteria | 0.89476947 | | 0.92432650 |
| ## Verrucomicrobia | 0.90620983 | | 0.93297623 |
| ## Armatimonadetes | 0.25579650 | | 0.30828409 |
| ## Planctomycetes | -0.09441804 | | -0.09806426 |
| ## Proteobacteria | 0.12659445 | | -0.03124084 |
| ## Bacteroidetes | 0.55391175 | | 0.45166935 |
| ## Deinococcus-Thermus | 0.07832152 | | 0.11399123 |
| ## Gemmatimonadetes | 0.53248997 | | 0.61152430 |
| ## | unclassified | Chloroflexi | Parcubacteria |
| ## Fusobacteria | -0.421289933 | -0.27807140 | -0.283216820 |
| ## Actinobacteria | -0.256547998 | -0.18032203 | -0.184214384 |
| ## Firmicutes | -0.463735784 | -0.36105188 | -0.353284937 |
| ## Acidobacteria | 0.938183100 | 0.91273061 | 0.894769467 |
| ## Candidatus_Saccharibacteria | 0.949849717 | 0.92724326 | 0.924326496 |
| ## unclassified | 1.000000000 | 0.97348566 | 0.965168777 |
| ## Chloroflexi | 0.973485655 | 1.00000000 | 0.986298372 |
| ## Parcubacteria | 0.965168777 | 0.98629837 | 1.000000000 |
| ## Verrucomicrobia | 0.977759885 | 0.99220269 | 0.995948122 |
| ## Armatimonadetes | 0.381925400 | 0.41771348 | 0.405919159 |
| ## Planctomycetes | -0.057795937 | -0.12639133 | -0.164933673 |
| ## Proteobacteria | 0.058866631 | -0.01150569 | -0.006459295 |
| ## Bacteroidetes | 0.551161910 | 0.38691522 | 0.367732760 |
| ## Deinococcus-Thermus | 0.003587177 | -0.15368927 | -0.148534166 |
| ## Gemmatimonadetes | 0.526809969 | 0.40981432 | 0.450818707 |
| ## | Verrucomicrobia | Armatimonadetes | Planctomycetes |
| ## Fusobacteria | -0.303504411 | 0.168525863 | 0.11673201 |
| ## Actinobacteria | -0.199703018 | -0.040094201 | 0.04730139 |
| ## Firmicutes | -0.383762989 | -0.177347837 | -0.32808762 |
| ## Acidobacteria | 0.906209829 | 0.255796504 | -0.09441804 |
| ## Candidatus_Saccharibacteria | 0.932976227 | 0.308284087 | -0.09806426 |
| ## unclassified | 0.977759885 | 0.381925400 | -0.05779594 |
| ## Chloroflexi | 0.992202693 | 0.417713478 | -0.12639133 |
| ## Parcubacteria | 0.995948122 | 0.405919159 | -0.16493367 |
| ## Verrucomicrobia | 1.000000000 | 0.404932208 | -0.13282189 |
| ## Armatimonadetes | 0.404932208 | 1.000000000 | 0.57522374 |
| ## Planctomycetes | -0.132821888 | 0.575223742 | 1.00000000 |
| ## Proteobacteria | 0.006981966 | 0.007270898 | 0.06327063 |
| ## Bacteroidetes | 0.420532471 | 0.049344296 | 0.29492361 |
| ## Deinococcus-Thermus | -0.113379200 | -0.349764842 | 0.24955112 |
| ## Gemmatimonadetes | 0.448273872 | -0.015339300 | 0.03750000 |
| ## | Proteobacteria | Bacteroidetes | Deinococcus-Thermus |
| ## Fusobacteria | -0.388015555 | -0.6488839 | -0.347897839 |
| ## Actinobacteria | -0.950486412 | -0.3314104 | -0.037058560 |
| ## Firmicutes | -0.702028717 | -0.6172501 | -0.477267605 |
| ## Acidobacteria | 0.126594453 | 0.5539118 | 0.078321521 |
| ## Candidatus_Saccharibacteria | -0.031240835 | 0.4516693 | 0.113991226 |
| ## unclassified | 0.058866631 | 0.5511619 | 0.003587177 |
| ## Chloroflexi | -0.011505694 | 0.3869152 | -0.153689265 |
| ## Parcubacteria | -0.006459295 | 0.3677328 | -0.148534166 |
| ## Verrucomicrobia | 0.006981966 | 0.4205325 | -0.113379200 |

```
## Armatimonadetes      0.007270898      0.0493443      -0.349764842
## Planctomycetes       0.063270633      0.2949236      0.249551121
## Proteobacteria       1.000000000      0.2248817      0.118933720
## Bacteroidetes        0.224881731      1.0000000      0.556154831
## Deinococcus-Thermus  0.118933720      0.5561548      1.000000000
## Gemmatimonadetes     0.282512065      0.3833599      0.606763948
##
## Gemmatimonadetes
## Fusobacteria         -0.4600615
## Actinobacteria       -0.2974225
## Firmicutes           -0.6147621
## Acidobacteria        0.5324900
## Candidatus_Saccharibacteria 0.6115243
## unclassified         0.5268100
## Chloroflexi          0.4098143
## Parcubacteria        0.4508187
## Verrucomicrobia      0.4482739
## Armatimonadetes     -0.0153393
## Planctomycetes       0.0375000
## Proteobacteria       0.2825121
## Bacteroidetes        0.3833599
## Deinococcus-Thermus  0.6067639
## Gemmatimonadetes     1.0000000
```

We then remove redundant data by removing the top triangle of the matrix.

```
# lower.tri selects the matrix' lower triangle, and we just remove all the data
gencor[lower.tri(gencor)] <- NA
```

Correlation map

We can now output a correlation map similar similar to a heatmap. Colors are set as white for no correlation and red and blue for positive and negative correlation respectively. We add the dendrogram obtained before above it.

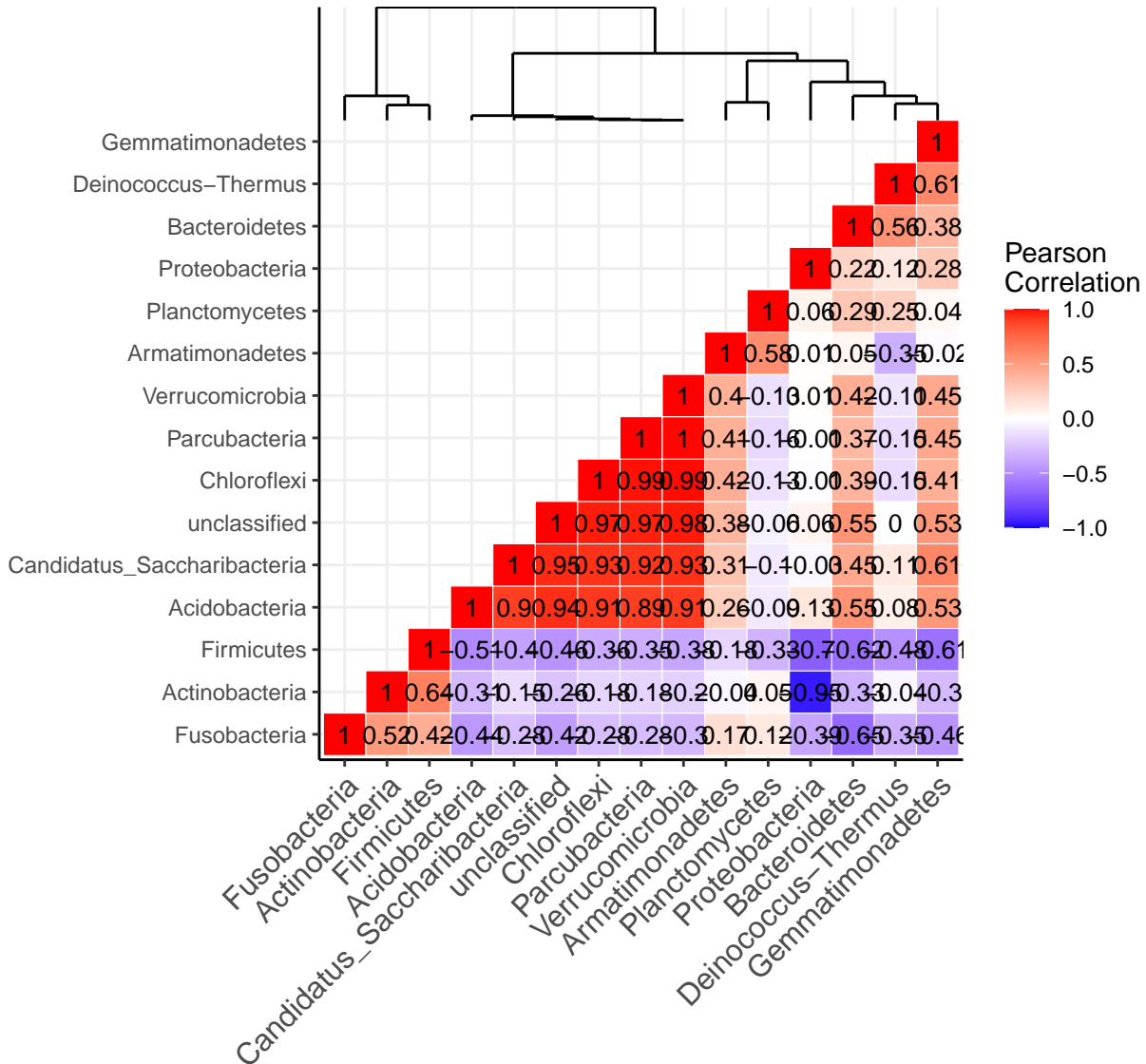
```
# use the ordered correlation matrix
cornmap <- gencor %>%
  # convert to dataframe for ggplot
  as.data.frame %>%
  # get the rownames to use for grouping
  rownames_to_column("grp") %>%
  # convert to long format, remove missing values (upper triangle)
  pivot_longer(-grp, names_to="key", values_to="v", values_drop_na=TRUE) %>%
  # make sure the order is preserved like before, otherwise elements may be shifted seemingly at random
  mutate(grp=factor(grp, levels=horder, ordered=TRUE),
    key=factor(key, levels=horder, ordered=TRUE)) %>%
  # pass it directly to ggplot; label will be the rounded correlation value within each cell
  ggplot(aes(key, grp, fill=v, label=round(v,2))) +
    # tilemap background
    geom_tile(color="white") +
    # set a blue to red gradient for [-1;1] correlation values
    scale_fill_gradient2(
      low="blue", high="red", mid="white",
      midpoint=0,
      limit=c(-1,1), space="Lab",
      name="Pearson\nCorrelation") +
```

```

# rotate x axis for easier viewing, remove axis title
theme(axis.text.x=element_text(angle=45, vjust=1, size=12, hjust=1),
      axis.title=element_blank()) +
# make sure we get a square matrix
coord_fixed() +
# add correlation values
geom_text(color="black", size=3.5) +
# draw the dendrogram; we offset y coordinates to place it above the tilemap
geom_segment(data=segment(dhcl), aes(x, y+nrow(gencor)+0.5, xend=xend, yend=yend+nrow(gencor)

```

cormap



```

# save it
ggsave("plots/cormap.pdf", cormap, width=12, height=10)

```

Principal component analysis

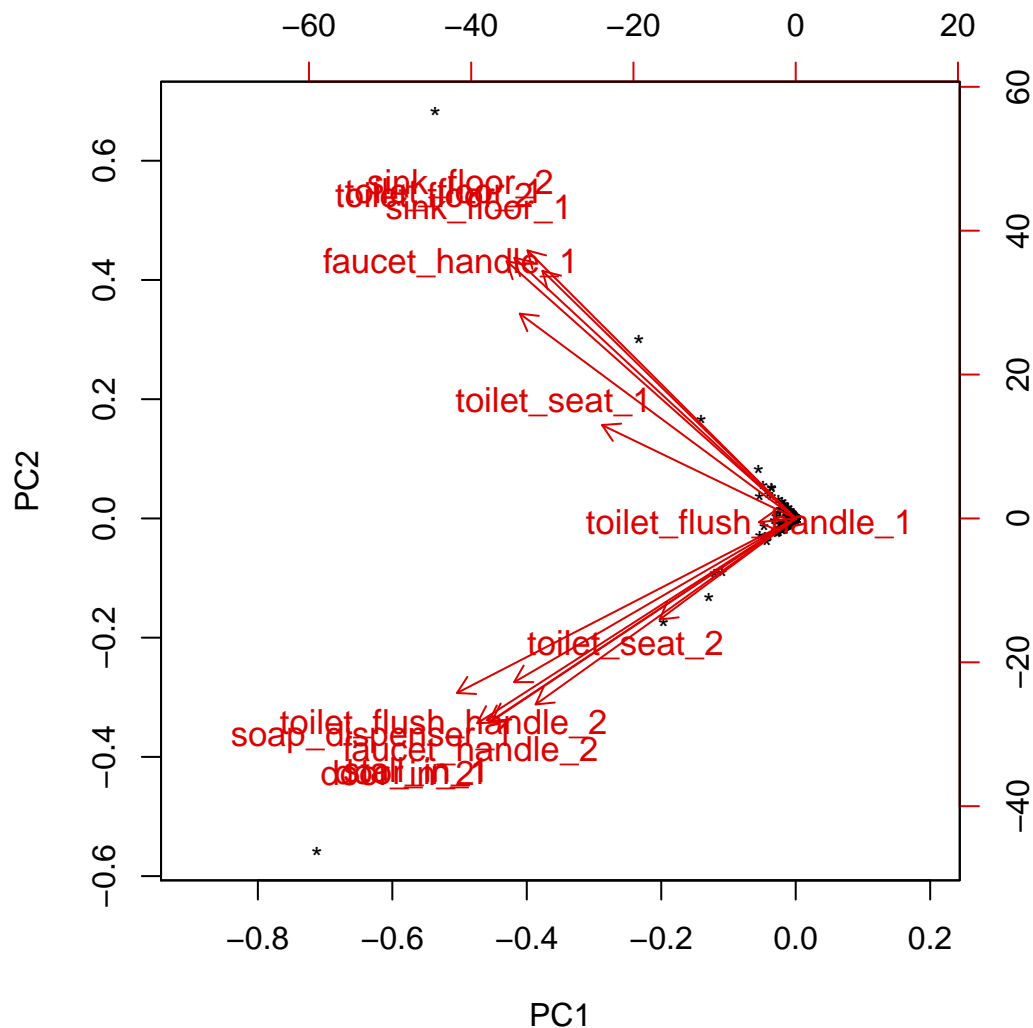
Finally, we can perform a simple PCA to evaluate if there is any grouping structure in the data.

```
# compute pca on numeric matrix, scale the variables to unit variance as recommended
pca <- prcomp(otumat, scale=TRUE)
# percent of variance explained by each axis
pca$sdev ^ 2 / sum(pca$sdev ^ 2)
```

```
## [1] 0.4057426457 0.2861426256 0.0988359817 0.0715174067 0.0565024752
## [6] 0.0254300512 0.0230228818 0.0113783480 0.0093346932 0.0063051507
## [11] 0.0021608007 0.0018565269 0.0011851263 0.0005852864
```

```
# 40.6% and 28.6% for PC1 and PC2
```

```
# render base graphics plot
biplot(pca,
  # draw arrows for each eigenvector
  var.axes=TRUE,
  xlab=rep("*", nrow(otumat)), cex=c(0.9,1.1),
  xlim=c(-0.9,0.2), expand=50,
  col=c("black", "#dd0000"))
```



```
pdf("plots/pca.pdf", 12, 10)
# plot the pca
biplot(pca,
       # draw arrows for each eigenvector
       var.axes=TRUE,
       xlab=rep("*", nrow(otumat)), cex=c(0.9,1.1),
       xlim=c(-0.9,0.2), expand=50,
       col=c("black", "#dd0000"))
dev.off()
```