Bacterial diversity in groundwater samples

Pierre-Yves Dupont

July 27, 2018

Table of Contents

## Libraries

library(phyloseq); packageVersion("phyloseq")

## [1] '1.26.1'

library(tidyverse)  
library(caching); packageVersion("caching")

## [1] '0.1.9'

library(cowplot)  
library(assertthat)  
library(ggplot2)  
library(RColorBrewer)  
library(colorRamps)  
library(ggrepel)  
require(car)  
library(ggpubr)  
library(Polychrome)

## Loading data

The objects computed using DADA2 have been saved using the caching library and can be loaded using the same library. Here we can load the phyloseq object ps and the data.frame extracted from it taxa.dt.

ps <- load.object("psc")$orig  
if(object.cached("taxa.dt")){  
 taxa.dt <- load.object("taxa.dt")  
} else{  
 taxa.dt <- psmelt(ps)  
 save.object(taxa.dt)  
}

## Loading Metadata

metadata <- read\_tsv("samples-metadata.tsv")  
metadata.df <- as.data.frame(metadata)  
rownames(metadata.df) <- metadata$SampleID  
sample\_data(ps) <- metadata.df  
if(!exists("taxa.dt")){  
 taxa.dt <- psmelt(ps)  
 save.object(taxa.dt)  
}

## Separate UC samples

uc\_samples <- c("GW180247","GW180419","GW180420","GW180421","GW180422","GW180423","GW180424","GW180425","GW180426","GW180427","GW180428", "GW1801011A", "GW180248")  
ps\_uc <- prune\_samples(sample\_data(ps)$SampleID %in% uc\_samples, ps)

## Diversity plots

### Filter low abundance taxa.

aggregation.level <- 'Genus'  
  
dt <- taxa.dt %>% filter(Blank == "No", SampleID %in% uc\_samples)  
top <- dt %>%   
 group\_by(!!(rlang::sym(aggregation.level))) %>%   
 summarise(Abundance=sum(Abundance)) %>%   
 mutate(AbundanceRel = Abundance/sum(Abundance)) %>%  
 filter(AbundanceRel >= 0.01)  
top <- top[[aggregation.level]]

### Compute relative abundance

dt.grouped <- dt %>%  
 filter(!!(rlang::sym(aggregation.level)) %in% top) %>%  
 mutate(!!(rlang::sym(aggregation.level)) := factor(!!(rlang::sym(aggregation.level)), levels=unique(!!(rlang::sym(aggregation.level))))) %>%  
 filter(!is.na(!!(rlang::sym(aggregation.level)))) %>% # Remove NA  
 group\_by(SampleID, !!(rlang::sym(aggregation.level))) %>%  
 summarize(Abundance = sum(Abundance)) %>%  
 mutate(Abundance = Abundance/sum(Abundance) \* 100) %>%  
 filter(!is.na(!!(rlang::sym(aggregation.level))))

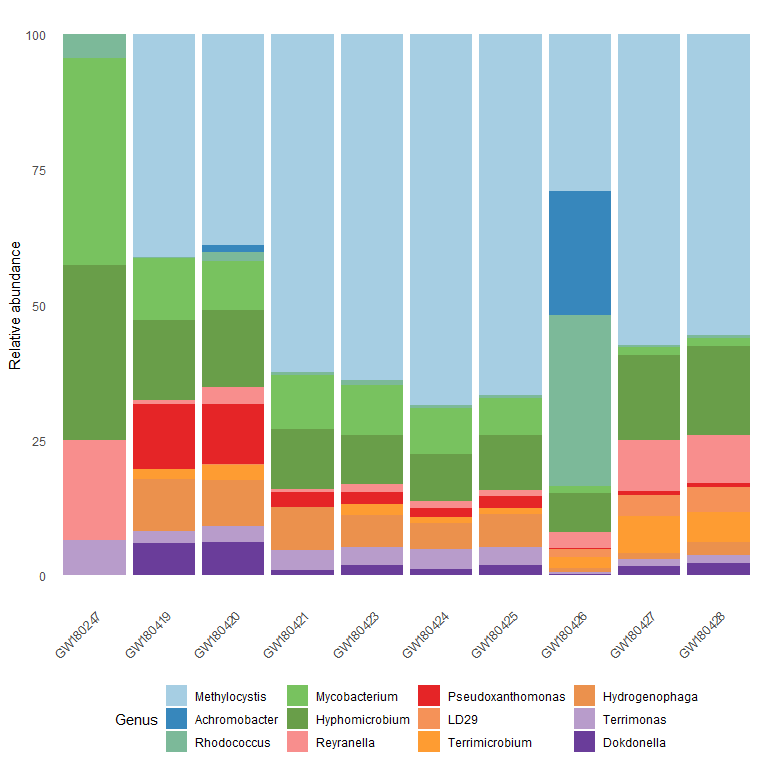
### Set the palette

colourCount <- length(unique(as.character(dt.grouped[[aggregation.level]])))  
palette <- colorRampPalette(brewer.pal(10,"Paired"))(colourCount)

### Plot

#### All samples

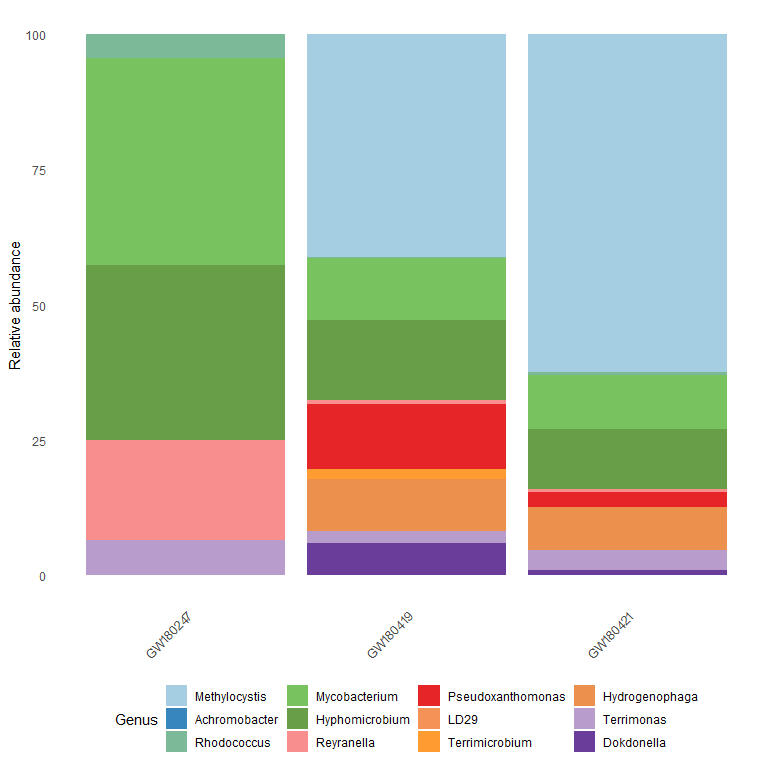
ggplot(dt.grouped, aes(SampleID, Abundance, fill = !!(rlang::sym(aggregation.level)))) +  
 geom\_bar(stat = "identity", position = "stack") +  
 # facet\_wrap(~SampleTypeAnalysed, ncol = 1, scales = "free\_x") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = palette) +  
 xlab(NULL) +  
 ylab("Relative abundance") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), panel.background = element\_blank()) +  
 theme(legend.position="bottom")



#### Only 3 samples

The following code builds a plot using the same taxa set as for the complete plot by subsetting the dt.grouped data.frame. This doesn’t change the distribution of the taxa.

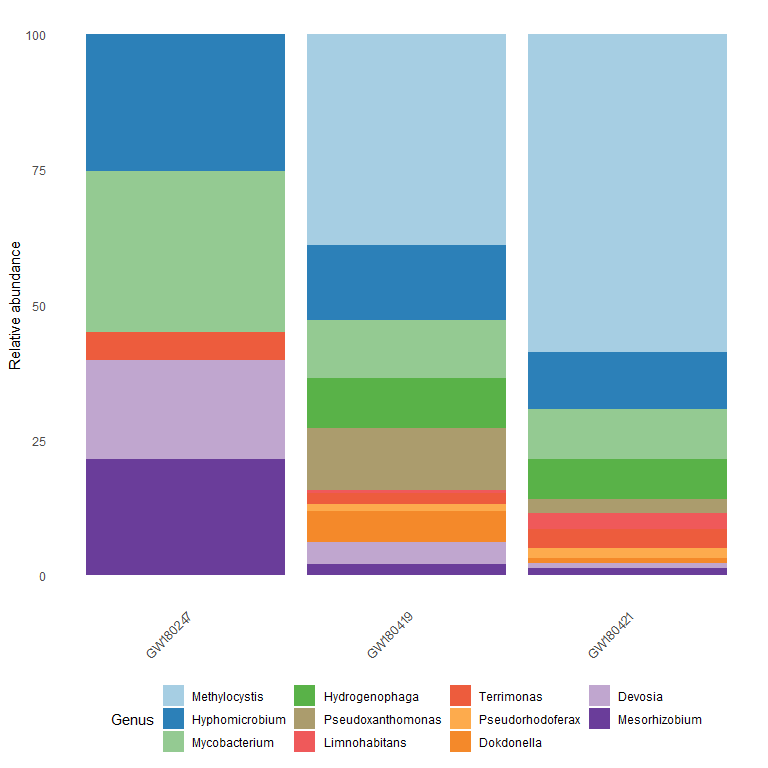
dt.grouped <- dt.grouped %>% filter(SampleID %in% c("GW180247", "GW180419", "GW180421"))  
ggplot(dt.grouped, aes(SampleID, Abundance, fill = !!(rlang::sym(aggregation.level)))) +  
 geom\_bar(stat = "identity", position = "stack") +  
 # facet\_wrap(~SampleTypeAnalysed, ncol = 1, scales = "free\_x") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = palette) +  
 xlab(NULL) +  
 ylab("Relative abundance") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), panel.background = element\_blank()) +  
 theme(legend.position="bottom")



The following code first subset the data then recomputes the taxa and build the abundance plot.

dt <- taxa.dt %>%  
 filter(Blank == "No", SampleID %in% uc\_samples) %>%   
 filter(SampleID %in% c("GW180247", "GW180419", "GW180421"))  
  
top <- dt %>%   
 group\_by(!!(rlang::sym(aggregation.level))) %>%   
 summarise(Abundance=sum(Abundance)) %>%   
 mutate(AbundanceRel = Abundance/sum(Abundance)) %>%  
 filter(AbundanceRel >= 0.01)  
top <- top[[aggregation.level]]  
  
dt.grouped <- dt %>%  
 filter(!!(rlang::sym(aggregation.level)) %in% top) %>%  
 mutate(!!(rlang::sym(aggregation.level)) := factor(!!(rlang::sym(aggregation.level)), levels=unique(!!(rlang::sym(aggregation.level))))) %>%  
 filter(!is.na(!!(rlang::sym(aggregation.level)))) %>% # Remove NA  
 group\_by(SampleID, !!(rlang::sym(aggregation.level))) %>%  
 summarize(Abundance = sum(Abundance)) %>%  
 mutate(Abundance = Abundance/sum(Abundance) \* 100) %>%  
 filter(!is.na(!!(rlang::sym(aggregation.level))))  
  
colourCount <- length(unique(as.character(dt.grouped[[aggregation.level]])))  
palette <- colorRampPalette(brewer.pal(10,"Paired"))(colourCount)

ggplot(dt.grouped, aes(SampleID, Abundance, fill = !!(rlang::sym(aggregation.level)))) +  
 geom\_bar(stat = "identity", position = "stack") +  
 # facet\_wrap(~SampleTypeAnalysed, ncol = 1, scales = "free\_x") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = palette) +  
 xlab(NULL) +  
 ylab("Relative abundance") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), panel.background = element\_blank()) +  
 theme(legend.position="bottom")



Both plots are correct. The first one corresponds to the distribution of the taxa identified as the most abundant in all the samples while the second corresponds to the distribution of the the most abundant taxa in these three samples. It is not surpising to see that you have such a large difference between the plots because your conditions (and likely your communities) are very different.

#### Top 5 taxa per sample

If you want to get the top N taxa per samples, you need to first group by your Samples and taxa then select the top\_n values based on the Abundance. You can change the N value if you want more or less taxa.

N <- 5  
dt <- taxa.dt %>%  
 filter(Blank == "No", SampleID %in% uc\_samples)  
  
dt.grouped <- dt %>%  
 mutate(!!(rlang::sym(aggregation.level)) := factor(!!(rlang::sym(aggregation.level)), levels=unique(!!(rlang::sym(aggregation.level))))) %>%  
 filter(!is.na(!!(rlang::sym(aggregation.level)))) %>% # Remove NA  
 group\_by(SampleID, !!(rlang::sym(aggregation.level))) %>%  
 summarize(Abundance = sum(Abundance)) %>%  
 mutate(Abundance = Abundance/sum(Abundance) \* 100) %>%  
 top\_n(5, Abundance) %>%   
 filter(!is.na(!!(rlang::sym(aggregation.level))))

I also use a different palette because there too many different values for the one that you were using (they are almost impossible to differentiate).

colourCount <- length(unique(as.character(dt.grouped[[aggregation.level]])))  
# palette <- colorRampPalette(brewer.pal(10,"Paired"))(colourCount)  
palette <- as.character(palette36.colors(colourCount))

The plot part doesn’t change.

ggplot(dt.grouped, aes(SampleID, Abundance, fill = !!(rlang::sym(aggregation.level)))) +  
 geom\_bar(stat = "identity", position = "stack") +  
 # facet\_wrap(~SampleTypeAnalysed, ncol = 1, scales = "free\_x") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = palette) +  
 xlab(NULL) +  
 ylab("Relative abundance") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), panel.background = element\_blank()) +  
 theme(legend.position="bottom")

