Mime-16s-emp

Clara Jégousse

3/3/2021

Table of Contents

# Load libraries

library(devtools)  
library(ggplot2)  
library(ggpubr)  
library(dada2)  
library(phyloseq)  
library(reshape2) # to use melt  
library(phylosmith)

# Visual setting

Import variables and functions for consistent plots.

source\_url("https://raw.githubusercontent.com/clarajegousse/mime-16s/main/scripts/visual-settings.r")

## ℹ SHA-1 hash of file is 3a57cdd05807ac4c92f1ab418ea362cb4ef8d917

# Load data

The results of dada2 sequence processing were organized into a phyloseq object containing all 1397 samples amplified with the EMP primers with metadata from Hafro.

ps <- readRDS("/Users/Clara/Projects/mime-16s/global-ps-emp.rds")  
  
dna <- Biostrings::DNAStringSet(taxa\_names(ps))  
names(dna) <- taxa\_names(ps)  
ps <- merge\_phyloseq(ps, dna)  
taxa\_names(ps) <- paste0("ASV", seq(ntaxa(ps)))  
ps

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 13417 taxa and 1397 samples ]  
## sample\_data() Sample Data: [ 1397 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 13417 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 13417 reference sequences ]

All metadata including measures from Hafro.

sample\_variables(ps)

## [1] "stn" "smp.num" "primer" "run"   
## [5] "stn.name" "stn.num" "cruise" "d2b"   
## [9] "year" "month" "day" "season"   
## [13] "lat" "lon" "depth.measured" "depth"   
## [17] "temp.avg" "salt.avg" "po4.avg" "sio2.avg"   
## [21] "no3.avg" "press" "chl.a" "phaeo"   
## [25] "rfsu" "filter.type" "transect" "date"   
## [29] "region" "iscar.nb" "zone"

rank\_names(ps)

## [1] "Kingdom" "Phylum" "Class" "Order" "Family" "Genus" "Species"

Number of taxa

ntaxa(ps)

## [1] 13417

# Deal with missing values in metadata with imputation of the mean of the two surrounding values

library(imputeTS)  
sample\_data(ps)$po4.avg <- round(na\_ma(sample\_data(ps)$po4.avg, k = 1), digits = 2)  
  
sample\_data(ps)$sio2.avg <- round(na\_ma(sample\_data(ps)$sio2.avg, k = 1), digits = 2)  
  
sample\_data(ps)$no3.avg <- round(na\_ma(sample\_data(ps)$no3.avg, k = 1), digits = 2)  
  
sample\_data(ps)[is.na(sample\_data(ps)$chl.a),]$chl.a <- 0  
sample\_data(ps)[is.na(sample\_data(ps)$phaeo),]$phaeo <- 0  
sample\_data(ps)[is.na(sample\_data(ps)$rfsu),]$rfsu <- 0

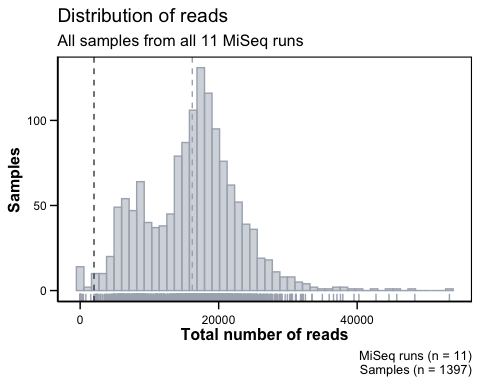
# Filtering

Filtering prevents spending time analyzing unreliable data, background noise (taxa that are actually just artifacts of the data collection process) and taxa that are seen rarely among samples.

## Filtering samples

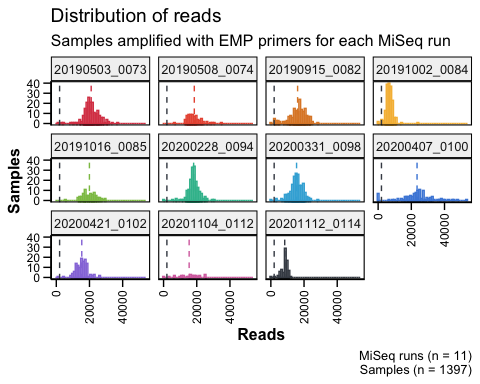
The objective is to remove samples with very low numbers of reads. So first we investigate the overall sequencing depths per sample setting a arbitrary threshold of 2000 reads as the minimum numbers of reads for a sample.

# number of reads per samples  
reads <- as.data.frame(sample\_sums(ps))  
colnames(reads) <- c("total")  
reads$run <- sample\_data(ps)$run  
reads$sample <- rownames(reads)  
  
gghistogram(reads, x = "total",  
 add = "mean", rug = TRUE,  
 bins = 50,  
 color = MediumGrey, fill = MediumGrey,  
 palette = Palette1) +  
 geom\_vline(xintercept = 2000, linetype = 2, col = DarkGrey) +  
 clean\_theme + theme(axis.text.x = element\_text(angle = 0, vjust = 0, hjust=.5))+  
 xlab("Total number of reads") + ylab("Samples") +  
 labs(title = "Distribution of reads",   
 subtitle = "All samples from all 11 MiSeq runs",   
 caption = paste0("MiSeq runs (n = ", length(unique(reads$run)), ")\n",  
 "Samples (n = ", dim(reads)[1], ")"))



We investigate the sequencing depth per samples for each MiSeq run.

gghistogram(reads, x = "total",  
 add = "mean", rug = TRUE,  
 color = "run", fill = "run",  
 bins = 50,  
 #color = MediumGrey, fill = MediumGrey,  
 palette = Palette1[-c(11,12)]) +   
 facet\_wrap(~run) +  
 geom\_vline(xintercept = 2000, linetype = 2, col = DarkGrey) +  
 clean\_theme + theme(legend.position = "none") +  
 xlab("Reads") + ylab("Samples") +  
 labs(title = "Distribution of reads",   
 subtitle = "Samples amplified with EMP primers for each MiSeq run",   
 caption = paste0("MiSeq runs (n = ", length(unique(reads$run)), ")\n",  
 "Samples (n = ", dim(reads)[1], ")"))



The plots confirm that we can filter out samples containing less than 2000 reads.

# based on the plots above define the minimum number of reads per sample  
min.reads <- 2000  
smp.keeper <- reads[reads$total >= min.reads,]$sample  
  
ps0 <- ps %>%  
 subset\_samples(rownames(sample\_data(ps)) %in% smp.keeper)  
ps0

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 13417 taxa and 1380 samples ]  
## sample\_data() Sample Data: [ 1380 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 13417 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 13417 reference sequences ]

The total number of samples removed because they contained less than 2000 reads.

length(reads[reads$total <= min.reads,]$sample)

## [1] 17

## Filter taxa

The samples were amplified with the EMP primers therefore it is reasonable to filter taxonomic features for which a high-rank taxonomy could not be assigned - like “Uncharacterized” at the Kingdom level. Such ambiguous features in this setting are almost always sequence artifacts that do not exist in nature. Here we remove everything that was not characterised as “Bacteria” at the kingdom level.

# because these were assigned with Silva  
ps0 <- subset\_taxa(ps0, Kingdom %in% c("Bacteria"))  
  
# check the phyla within Bacteria  
table(tax\_table(ps0)[, "Phylum"], exclude = NULL)

##   
## Acidobacteriota Actinobacteriota   
## 168 329   
## AncK6 Armatimonadota   
## 5 5   
## Bacteroidota Bdellovibrionota   
## 1799 328   
## Caldisericota Calditrichota   
## 1 4   
## Campilobacterota Chloroflexi   
## 47 358   
## Cyanobacteria Dadabacteria   
## 1110 20   
## Deinococcota Dependentiae   
## 12 20   
## Desulfobacterota Fibrobacterota   
## 152 13   
## Firmicutes Fusobacteriota   
## 165 17   
## Gemmatimonadota Hydrogenedentes   
## 62 13   
## Latescibacterota Margulisbacteria   
## 10 127   
## Marinimicrobia (SAR406 clade) MBNT15   
## 414 3   
## Methylomirabilota Myxococcota   
## 2 139   
## NB1-j Nitrospinota   
## 68 99   
## Nitrospirota Patescibacteria   
## 15 66   
## PAUC34f Planctomycetota   
## 58 641   
## Poribacteria Proteobacteria   
## 7 5141   
## SAR324 clade(Marine group B) Schekmanbacteria   
## 67 4   
## Spirochaetota Sva0485   
## 12 2   
## Thermotogota Verrucomicrobiota   
## 1 549   
## WS2 <NA>   
## 1 683

## Prevalence filtering

Prevalence filtering is unsupervised, relying only on the data in this experiment, and a parameter that we choose after exploring the data. Thus, this filtering step can be applied even in settings where taxonomic annotation is unavailable or unreliable.

First, explore the relationship of prevalence and total read count for each feature. Sometimes this reveals outliers that should probably be removed, and also provides insight into the ranges of either feature that might be useful.

Define prevalence of each taxa (in how many samples did each taxa appear at least once).

# Define prevalence of each taxa  
# (in how many samples did each taxa appear at least once)  
prev0 = apply(X = otu\_table(ps0),  
 MARGIN = ifelse(taxa\_are\_rows(ps0), yes = 1, no = 2),  
 FUN = function(x){sum(x > 0)})  
prevdf = data.frame(Prevalence = prev0,  
 TotalAbundance = taxa\_sums(ps0),  
 tax\_table(ps0))  
  
# Define prevalence threshold as 1% of total samples  
prevalenceThreshold = round(0.01 \* nsamples(ps0), digits = 0)  
prevalenceThreshold

## [1] 14

# Execute prevalence filter, using `prune\_taxa()` function  
ps1 = prune\_taxa((prev0 > prevalenceThreshold), ps0)  
ps1

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 1753 taxa and 1380 samples ]  
## sample\_data() Sample Data: [ 1380 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 1753 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 1753 reference sequences ]

table(prevdf$Phylum)

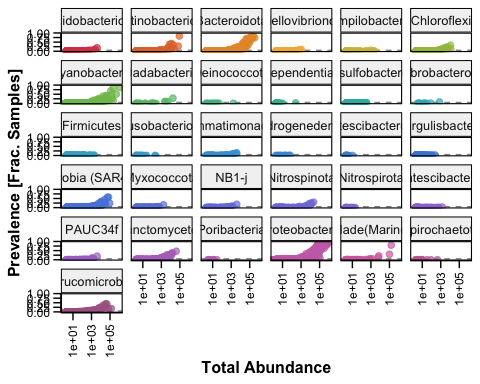
##   
## Acidobacteriota Actinobacteriota   
## 168 329   
## AncK6 Armatimonadota   
## 5 5   
## Bacteroidota Bdellovibrionota   
## 1799 328   
## Caldisericota Calditrichota   
## 1 4   
## Campilobacterota Chloroflexi   
## 47 358   
## Cyanobacteria Dadabacteria   
## 1110 20   
## Deinococcota Dependentiae   
## 12 20   
## Desulfobacterota Fibrobacterota   
## 152 13   
## Firmicutes Fusobacteriota   
## 165 17   
## Gemmatimonadota Hydrogenedentes   
## 62 13   
## Latescibacterota Margulisbacteria   
## 10 127   
## Marinimicrobia (SAR406 clade) MBNT15   
## 414 3   
## Methylomirabilota Myxococcota   
## 2 139   
## NB1-j Nitrospinota   
## 68 99   
## Nitrospirota Patescibacteria   
## 15 66   
## PAUC34f Planctomycetota   
## 58 641   
## Poribacteria Proteobacteria   
## 7 5141   
## SAR324 clade(Marine group B) Schekmanbacteria   
## 67 4   
## Spirochaetota Sva0485   
## 12 2   
## Thermotogota Verrucomicrobiota   
## 1 549   
## WS2   
## 1

keepPhyla = table(prevdf$Phylum)[(table(prevdf$Phylum) > 5)]  
prevdf1 = subset(prevdf, Phylum %in% names(keepPhyla))  
  
# Filter entries with unidentified Phylum.  
ps2 = subset\_taxa(ps1, Phylum %in% names(keepPhyla))  
ps2

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 1701 taxa and 1380 samples ]  
## sample\_data() Sample Data: [ 1380 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 1701 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 1701 reference sequences ]

ggplot(prevdf1, aes(TotalAbundance, Prevalence / nsamples(ps0), color=Phylum)) +  
 geom\_hline(yintercept = 0.05, alpha = 0.5, linetype = 2) +   
 geom\_point(size = 2, alpha = 0.7) +  
 scale\_x\_log10() +   
 tax\_color\_scale(ps0, "Phylum") +  
 xlab("Total Abundance") + ylab("Prevalence [Frac. Samples]") +  
 facet\_wrap(~Phylum) + clean\_theme + theme(legend.position="none")

## Warning: Transformation introduced infinite values in continuous x-axis



## Agglomerate taxa at the Genus level

There is a lot of species, sub-species, or strains with functional redundancy in the marine microbial community, so we can agglomerate the data features corresponding to closely related taxa (here at the Genus level) as we looking at overall patterns.

taxGlomRank = "Genus"  
length(get\_taxa\_unique(ps2, taxonomic.rank = taxGlomRank))

## [1] 181

ps3 = tax\_glom(ps2, taxrank = taxGlomRank)

## Filter specific samples

For now, let’s focus on one survey.

# ps4 <- subset\_samples(ps3, is.na(cruise) == FALSE & cruise != "B8-2010")  
ps4 <- subset\_samples(ps3, cruise == "B7-2017")  
ps4

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 180 taxa and 82 samples ]  
## sample\_data() Sample Data: [ 82 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 180 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 180 reference sequences ]

# Data normalisation

## Normalising OTU abundance

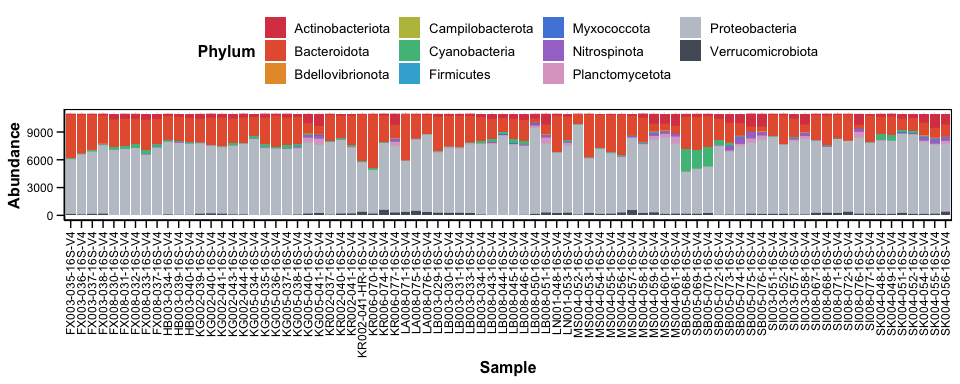
Normalize number of reads in each sample using median sequencing depth (cf. Daniel Vaulot).

# with microbiomeSeq  
# ps4n <- normalise\_data(ps4, norm.method = "relative")  
# phylosmith  
# ps4n <- relative\_abundance(ps4)  
  
total = median(sample\_sums(ps4))  
standf = function(x, t=total) round(t \* (x / sum(x)))  
ps4n = transform\_sample\_counts(ps4, standf)  
ps4n

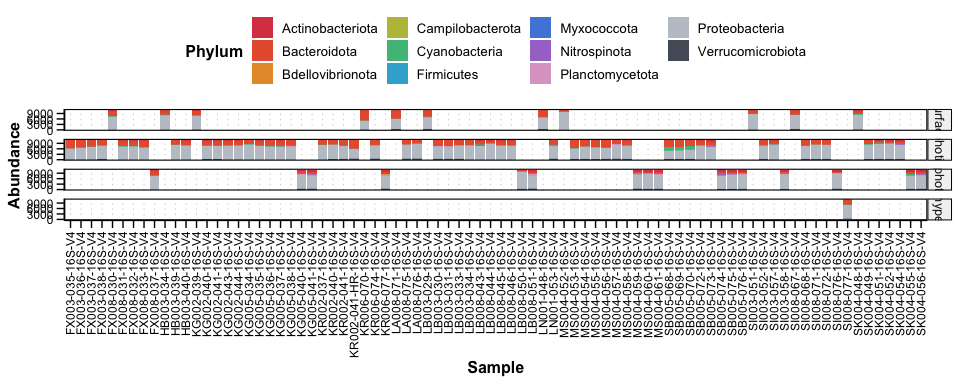
## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 180 taxa and 82 samples ]  
## sample\_data() Sample Data: [ 82 samples by 31 sample variables ]  
## tax\_table() Taxonomy Table: [ 180 taxa by 7 taxonomic ranks ]  
## refseq() DNAStringSet: [ 180 reference sequences ]

# Beta diversity

plot\_bar2(ps4n, x = "Sample", y = "Abundance", fill = "Phylum")



plot\_bar2(ps4n, x = "Sample", y = "Abundance", fill = "Phylum") + facet\_grid(zone~.)

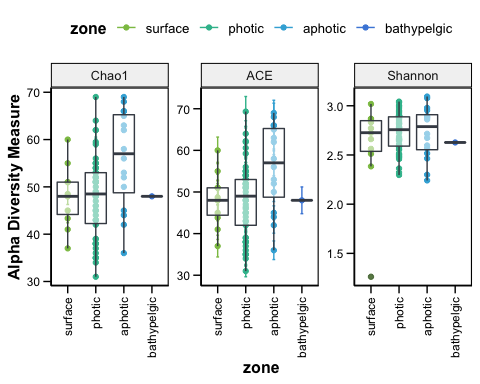


# Alpha diversity

TODO: ANOVA to confirm is alpha diversity is significantly different between zones

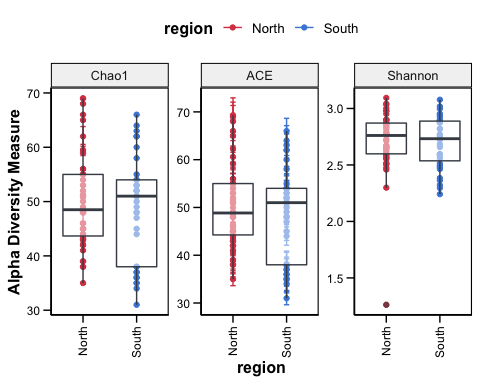
## Photic zone

p <- plot\_richness(ps4n, measures = c("Chao1", "Shannon", "ACE"),   
 x = "zone", color = "zone") +  
 scale\_color\_manual(values = c(Grass, Mint, Aqua, Jeans))  
  
p + geom\_boxplot(data = p$data, aes(x = zone, y = value), color = DarkGrey,   
 alpha = 0.5) + clean\_theme



## Region

p <- plot\_richness(ps4n, measures = c("Chao1", "Shannon", "ACE"),   
 x = "region", color = "region") +  
 scale\_color\_manual(values = c(Grapefruit, Jeans))  
  
p + geom\_boxplot(data = p$data, aes(x = region, y = value), color = DarkGrey,   
 alpha = 0.5) + clean\_theme

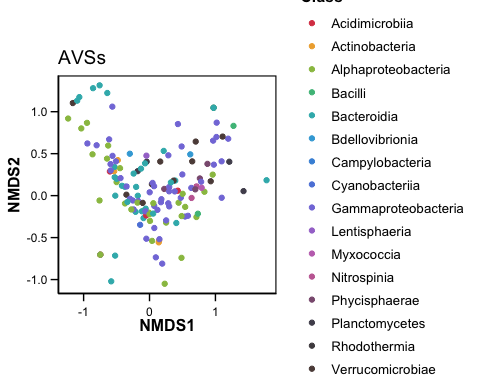


# Ordination

ps4n.ord <- ordinate(ps4n, "NMDS", "bray")

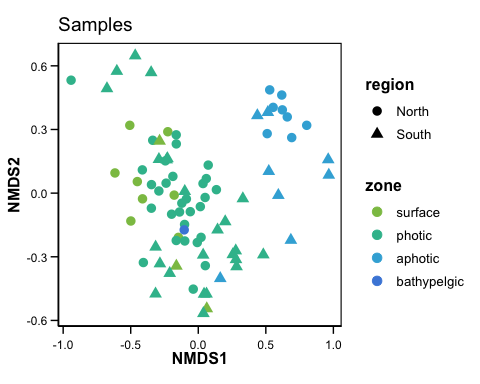
## Display ASVs

plot\_ordination(ps4n, ps4n.ord, type = "taxa",   
 color = "Class", title = "AVSs") +  
 tax\_color\_scale(ps4, "Class") +  
 clean\_theme + theme(aspect.ratio = 1,  
 axis.text.x = element\_text(angle = 0,   
 hjust = .5,   
 vjust = 0),  
 legend.position = "right")



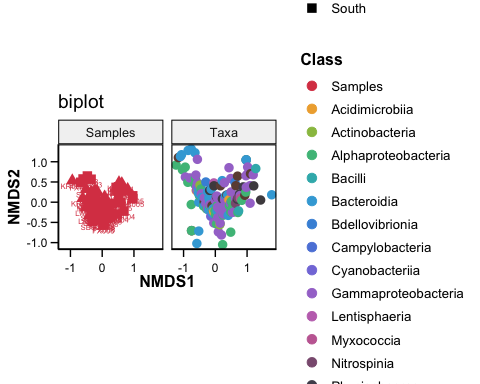
## Display samples

plot\_ordination(ps4n, ps4n.ord, type="samples", color="zone",   
 shape="region", title="Samples") + geom\_point(size=3) +  
 scale\_color\_manual(values = c(Grass, Mint, Aqua, Jeans)) +  
 clean\_theme + theme(aspect.ratio = 1,  
 axis.text.x = element\_text(angle = 0,   
 hjust = .5,   
 vjust = 0),  
 legend.position = "right")



## AVSs and samples

plot\_ordination(ps4n, ps4n.ord, type="split", color = "Class",   
 shape = "region", title="biplot", label = "stn") +   
 geom\_point(size=3) +   
 tax\_color\_scale(ps4, "Class") +  
 clean\_theme + theme(aspect.ratio = 1,  
 axis.text.x = element\_text(angle = 0,   
 hjust = .5,   
 vjust = 0),  
 legend.position = "right")



# Correlation matrix