

Sequence data input and subsetting

Stijn Schreven

4 March 2021

Contents

Introduction	1
Load packages	2
1. Load input files and create a phyloseq object	2
2. Removing mitochondrial and chloroplast DNA	3
3. Decontaminate dataset	3
3.1. Input data	3
3.2. Plot for-loop	3
3.3. Subset contaminants from taxonomic table	5
3.4. Calculate % of total reads per contaminant ASV	5
3.5. Decontaminate dataset	5
4. Subsetting	6
4.1. From total dataset ps1 (incl. contaminants)	6
4.2. From decontaminated dataset ps1.decontam	6
4.3. Exclude low-quality samples	7
5. Saving subsets in compressed RDS format	8

Introduction

This markdown file creates the *phyloseq* object that is required for all downstream analysis on the sequencing data.

We go through several filter steps:

- excluding mitochondrial and chloroplast reads;
- identifying and excluding contaminant reads;
- excluding low-quality samples, based on qPCR and HiSeq thresholds.

Load packages

```
library(phyloseq)
library(microbiome)
library(ggplot2)
library(plyr)
library(ape)
library(ggpubr)
```

1. Load input files and create a phyloseq object

In the mapping file and manuscript, treatment codes are:

- S/E = untreated substrate, untreated eggs;
- Si/E = sterilized substrate with 10% untreated substrate, untreated eggs;
- Si/Es = sterilized substrate with 10% untreated substrate, sterilized eggs;
- Ss/E = sterilized substrate, untreated eggs.

Besides, the complete dataset includes four samples of a preliminary experiment, in which we had the treatment Ss/Es = sterilized substrate and sterilized eggs. This treatment was not included in the main experiment in the publication.

```
ps <- read_phyloseq(otu.file = "./input_data/Schreven_Ch4_seqdata.biom1",
                    taxonomy.file = NULL,
                    metadata.file = "./input_data/Schreven_Ch4_mapping_file.csv",
                    type = "biom")
```

```
## Time to complete depends on OTU file size
```

```
# tree file
treefile_p1 <- read.tree("./input_data/Schreven_Ch4_seqdata_tree.tre")

# merge tree into phyloseq
ps <- merge_phyloseq(ps, treefile_p1)
ps
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 4584 taxa and 210 samples ]
## sample_data() Sample Data: [ 210 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 4584 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 4584 tips and 4583 internal nodes ]
```

```
# order factor levels
ps.meta <- meta(ps)
ps.meta$Type <- factor(ps.meta$Type, levels(ps.meta$Type)[c(6,3,1,2,5,4)])
sample_data(ps) <- ps.meta
```

2. Removing mitochondrial and chloroplast DNA

```
ps1 <- subset_taxa(ps, Family != "f__Mitochondria")
ps1 <- subset_taxa(ps1, Order != "o__Chloroplast")
```

3. Decontaminate dataset

Visual contaminant identification: by inspecting the correlation between DNA concentration per sample and the number of reads per ASV per sample. Those ASVs with a negative correlation (and present in at least 4 samples) are identified as likely contaminants.

3.1. Input data

```
# remove samples of 0 reads
psx <- prune_samples(sample_sums(ps1) > 0, ps1)
psx.df <- meta(psx) # meta data
psx.df$LibrarySize <- sample_sums(psx)
psx.df <- psx.df[order(psx.df$LibrarySize),] # order by library size (# reads per sample).
psx.df$Index <- seq(nrow(psx.df)) # index of samples based on rank by library size.

# file with samples, librarysize, DNAconc
psz1 <- psx.df[, c("Description", "DNAconc", "LibrarySize")]
# file with samples, taxa, reads per sample/taxon
psz2 <- as.data.frame(t(abundances(psx)))
psz2$Description <- rownames(psz2)
psz2.m <- reshape2::melt(psz2)
```

Using Description as id variables

```
colnames(psz2.m) <- c("Description", "OTU", "nReads")
# merge files
psz <- base::merge(psz2.m, psz1, by = "Description")
psz$freq <- psz$nReads / psz$LibrarySize
```

3.2. Plot for-loop

Note: this code chunk creates a PDF file with 4419 pages, the code takes quite some runtime (10-15 minutes on 16GB RAM drive)! Code in for-loop to create 1 PDF with all plotted correlations. Source: <https://stackoverflow.com/questions/26034177/save-multiple-ggplots-using-a-for-loop>

```
plot_list = list()

for (i in 1:nlevels(psz$OTU)) {
  p = ggplot(psz[psz$OTU %in% levels(psz$OTU)[i],],
    aes(x=DNAconc, y=freq)) +
    geom_point(size=3) +
    labs(x="[DNA]", y="rel. abundance") +
```

```

    ggtitle(levels(psz$OTU)[i]) +
    geom_smooth(method="lm") +
    scale_y_continuous(trans="log10")
  plot_list[[i]] = p
}

# create the pdf file, each page is a separate plot.
pdf("./figures/Schreven_Ch4_ASV_correlations_RelAbd_DNA.pdf")
for (i in 1:nlevels(psz$OTU)) {
  print(plot_list[[i]])
}
dev.off()

```

```

## pdf
## 2

```

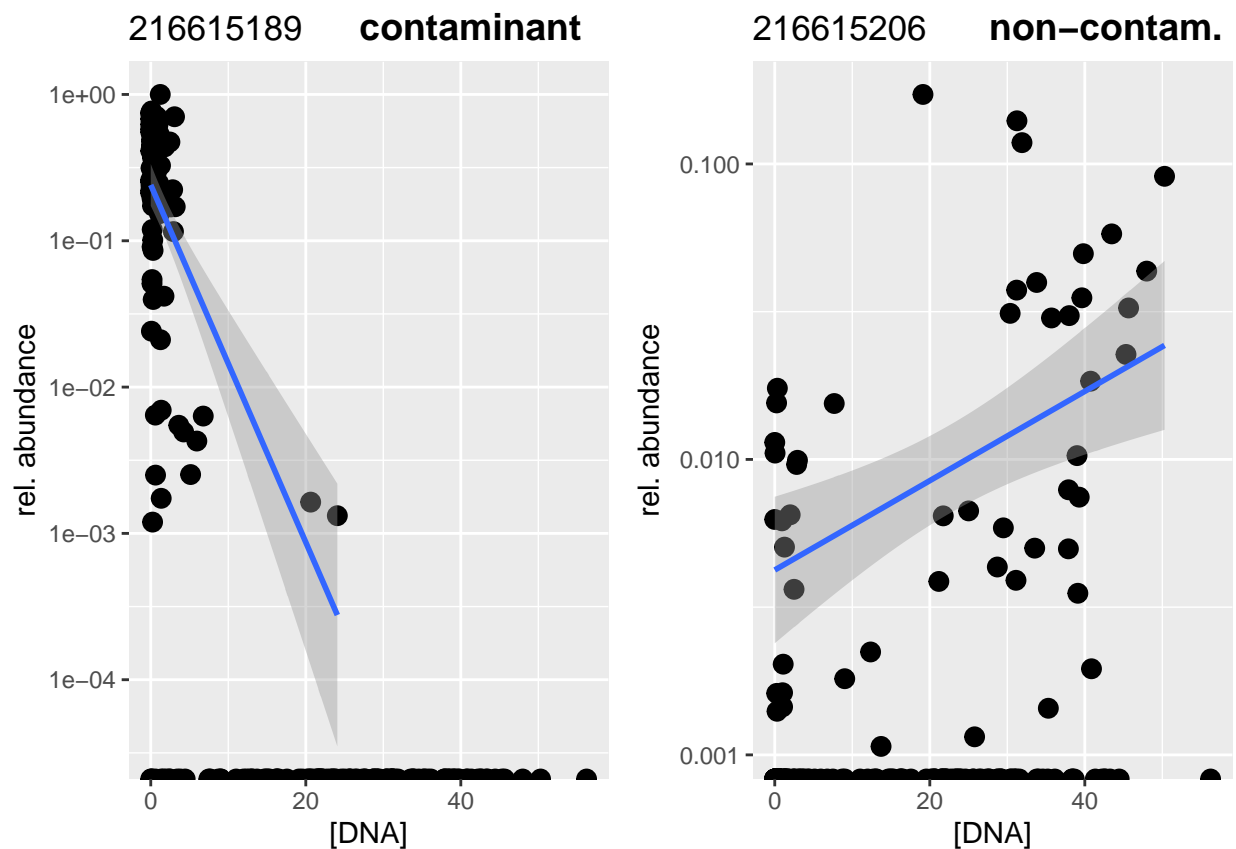
Now, visually inspect the pdf file (all 4419 pages!) and note down the page numbers of likely contaminants in Text file `./tables/Schreven_Ch4_contaminant_OTU_by_plot.txt`.

Example of a contaminant and a non-contaminant OTU:

```

ggarrange(plotlist = plot_list[c(1043,635)], ncol = 2, align = "hv",
  labels = c("contaminant", "non-contam."),
  label.x = .4, label.y = 1)

```



```
# Result: 188 contaminant OTUs
visContam <- read.delim("./input_data/Schreven_Ch4_contaminant_OTU_by_plot.txt")
# numbers are pdf pages, corresponding with the row index in psx otu_table.
```

3.3. Subset contaminants from taxonomic table

```
# extract tax_table
psx.tax <- as.data.frame(psx@tax_table)
psx.tax$OTU <- rownames(psx.tax)
tax_table(psx) <- tax_table(as.matrix(psx.tax))

rownames(psx.tax) <- NULL # reset rownames for subset based on index
psx.tax$contam <- rownames(psx.tax) %in% visContam$OTU
table(psx.tax$contam) # works: 188 OTUs T, rest F

##
## FALSE TRUE
## 4231 188
```

```
psx.tax$reads <- taxa_sums(psx)
rownames(psx.tax) <- psx.tax$OTU # restore OTU as rownames
```

3.4. Calculate % of total reads per contaminant ASV

Supplementary Table 4.

```
# create table of % reads per contaminant OTU
contam.otu2 <- subset(psx.tax, contam == "TRUE")
contam.otu2$freq <- contam.otu2$reads / sum(sample_sums(ps1))

contam.sum <- ddply(contam.otu2, ~ Domain + Phylum + Class + Order +
  Family + Genus, summarise,
  "number of ASVs" = length(OTU),
  "number of reads" = sum(reads),
  "% of total reads" = sum(freq))
contam.sum <- contam.sum[order(contam.sum$'number of reads', decreasing = T),]

# export table
write.csv(contam.sum, "./tables/Supplementary_Table_S4.csv")
```

3.5. Decontaminate dataset

Remove the OTUs that have been identified as contaminants.

```
ps1.decontam <- prune_taxa(!psx.tax$contam, ps1)
ps1.decontam <- prune_samples(sample_sums(ps1.decontam) > 0, ps1.decontam)
ps1.decontam
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 4231 taxa and 208 samples ]
## sample_data() Sample Data: [ 208 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 4231 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 4231 tips and 4230 internal nodes ]
```

```
ntaxa(ps1) - ntaxa(ps1.decontam) # 188 OTUs removed.
```

```
## [1] 188
```

```
sum(abundances(ps1)) - sum(abundances(ps1.decontam)) # 784748 reads removed.
```

```
## [1] 784748
```

4. Subsetting

4.1. From total dataset ps1 (incl. contaminants)

```
# negative controls
ps1.contr <- subset_samples(ps1, Type == "isol_blank" | Type == "pcr_blank")
ps1.contr <- prune_taxa(taxa_sums(otu_table(ps1.contr)) > 0, ps1.contr)
ps1.contr
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 549 taxa and 16 samples ]
## sample_data() Sample Data: [ 16 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 549 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 549 tips and 548 internal nodes ]
```

4.2. From decontaminated dataset ps1.decontam

```
# mocks
ps1.mock <- subset_samples(ps1.decontam, Type == "mock")
ps1.mock <- prune_taxa(taxa_sums(otu_table(ps1.mock)) > 0, ps1.mock)
ps1.mock
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 100 taxa and 8 samples ]
## sample_data() Sample Data: [ 8 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 100 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 100 tips and 99 internal nodes ]
```

```
# technical replicates of DNA extraction, 27 samples
ps1.biol <- subset_samples(ps1.decontam, Description %in% c(
  "15.N", "15.N1", "15.N2", "15.N3",
  "30.N", "30.N1", "30.N2", "30.N3",
```

```

"32.K", "32.K1", "32.K2", "32.K3",
"18.K", "18.K1", "18.K2", "18.K3",
"23.K", "23.K1", "23.K2", "23.K3",
"15.K", "15.K1", "15.K2", "15.K3",
"4.K", "4.K1", "4.K2"))
ps1.biol <- prune_taxa(taxa_sums(otu_table(ps1.biol)) > 0, ps1.biol)
ps1.biol

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 662 taxa and 27 samples ]
## sample_data() Sample Data: [ 27 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 662 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 662 tips and 661 internal nodes ]

```

```

# technical replicates of PCR, 18 samples
ps1.tech <- subset_samples(ps1.decontam, Description %in% c(
  "18.N", "18.N.Tdup",
  "L7", "L7.Tdup",
  "L18", "L18.Tdup",
  "15.N", "15.N.Tdup",
  "33.M", "33.M.Tdup",
  "16.K", "16.K.Tdup",
  "L3", "L3.Tdup1", "L3.Tdup2",
  "4.K1", "4.K1.Tdup1", "4.K1.Tdup2"))
ps1.tech <- prune_taxa(taxa_sums(otu_table(ps1.tech)) > 0, ps1.tech)
ps1.tech

```

```

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 643 taxa and 18 samples ]
## sample_data() Sample Data: [ 18 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 643 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 643 tips and 642 internal nodes ]

# full experiment (excluding duplicates)
ps1.exp <- subset_samples(ps1.decontam, Type == "larvae" | Type == "substrate")
ps1.exp <- subset_samples(ps1.exp, Pilot == "no")
ps1.exp <- subset_samples(ps1.exp, Duplicate == "no")
ps1.exp <- prune_taxa(taxa_sums(otu_table(ps1.exp)) > 0, ps1.exp)
ps1.exp

```

```

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2834 taxa and 127 samples ]
## sample_data() Sample Data: [ 127 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 2834 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2834 tips and 2833 internal nodes ]

```

4.3. Exclude low-quality samples

Excluding all low-quality samples, implies:

- excluding qPCR samples that scored negative (within 5 cycles difference of lowest NTC);

- and excluding low-read samples (fewer than 5000 reads).

In addition, we excluded:

- samples of $t = 22$, since there is no reason to include (no valid comparisons with other treatments to make). It can be included in the performance data Figures, but also there it should be excluded from statistics;
- samples 26.K and 39.M, because they have very likely been mixed up in the lab work (derived from ordinations), they were adjacent G11 and G12 in library G;
- samples of containers 6 and 7 because of green fungal overgrowth in container, most likely contamination and not originating from eggs.

```
# samples to exclude based on qPCR output
qpcr_excl2 <- read.delim("./input_data/Schreven_Ch4_qPCR_excl_nodoubt.txt")
# 45 samples

# subset for workable dataset
ps1.work <- subset_samples(ps1.exp, !sample_names(ps1.exp) %in% qpcr_excl2$qPCR_excl)
ps1.work <- subset_samples(ps1.work, sample_sums(ps1.work) > 5000)
ps1.work <- subset_samples(ps1.work, Timepoint != 22)
ps1.work <- subset_samples(ps1.work, ! sample_names(ps1.work) %in% c("26.K", "39.M"))
ps1.work <- subset_samples(ps1.work, ! sample_names(ps1.work) %in% c(
  "6.K", "6.M", "6.N", "7.K", "7.M", "7.N"))
ps1.work <- prune_taxa(taxa_sums(otu_table(ps1.work)) > 0, ps1.work)
ps1.work
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2222 taxa and 93 samples ]
## sample_data() Sample Data: [ 93 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 2222 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2222 tips and 2221 internal nodes ]
```

5. Saving subsets in compressed RDS format

```
# raw and filtered datasets
saveRDS(ps, "./phyobjects/ps.rds") # all reads
saveRDS(ps1, "./phyobjects/ps1.rds") # excl. mitochondria and chloroplasts
saveRDS(ps1.decontam, "./phyobjects/ps1.decontam.rds") # excl. contaminants

# quality control
saveRDS(ps1.contr, "./phyobjects/ps1.contr.rds") # no-template controls
saveRDS(ps1.mock, "./phyobjects/ps1.mock.rds") # positive controls
saveRDS(ps1.biol, "./phyobjects/ps1.biol.rds") # DNA isolation replicates
saveRDS(ps1.tech, "./phyobjects/ps1.tech.rds") # PCR replicates

# experimental data
saveRDS(ps1.exp, "./phyobjects/ps1.exp.rds")
saveRDS(ps1.work, "./phyobjects/ps1.work.rds") # excl. low-quality samples
```