

Sequence data input and subsetting

Stijn Schreven

23 April 2022

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. Prepare data	3
3.2. Sequencing depth per sample	4
3.3. Correlation plots	5
3.4. Plot % contaminant reads per sample	6
3.5. Table and plot of contaminant ASVs	8
3.6. Decontaminating the dataset	11
4. Subsetting	11
4.1. From total dataset ps1 (including contaminants)	11
4.2. From decontaminated dataset ps1.decontam	11
5. Saving datasets in compressed RDS format	13

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(ape)
library(reshape2)
library(plyr)
library(ggplot2)
library(ggrepel)
library(ggpubr)
```

1. Load input files and create a phyloseq object

Loading the five sequence libraries of the study (350 samples). The dataset contains all sequencing runs, run through NGTax 1.0 and SILVA database version 128.

```
# create phyloseq object from biom1 and mapping files
psconE <- read_phyloseq(
  otu.file = "./input_data/Schreven_libAE_seqdata.biom1",
  taxonomy.file = NULL,
  metadata.file = "./input_data/Schreven_libAE_mappingfile.csv",
  type = "biom")
```

Time to complete depends on OTU file size

```
# load tree file
treefile_p1conE <- read.tree("./input_data/Schreven_libAE_allotus.tre")

# merge tree into phyloseq
psconE <- merge_phyloseq(psconE, treefile_p1conE)
psconE
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2972 taxa and 350 samples ]
## sample_data() Sample Data: [ 350 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 2972 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2972 tips and 2971 internal nodes ]
```

```
# reorder factor levels in metadata
psconE.meta <- meta(psconE)
psconE.meta$Diet <- factor(psconE.meta$Diet, levels(psconE.meta$Diet)[c(1, 3, 2, 4)])
psconE.meta$Density <- factor(psconE.meta$Density,
  levels(psconE.meta$Density)[c(1, 4, 2, 3, 5)])
psconE.meta$Timepoint <- factor(psconE.meta$Timepoint,
  levels(psconE.meta$Timepoint)[c(1, 4, 2, 3, 5)])
psconE.meta$Type <- factor(psconE.meta$Type,
  levels(psconE.meta$Type)[c(6, 4, 2, 1, 3, 5)])

# update metadata in psconE phyloseq object with reordered factors
sample_data(psconE) <- psconE.meta
```

2. Removing mitochondrial and chloroplast DNA

```
ps1 <- subset_taxa(psconE, Family != "f__Mitochondria")
ps1 <- subset_taxa(ps1, Class != "c__Chloroplast")
ps1

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2814 taxa and 350 samples ]
## sample_data() Sample Data: [ 350 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 2814 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2814 tips and 2813 internal nodes ]

ntaxa(psconE)-ntaxa(ps1) # 158 otus removed

## [1] 158
```

3. Decontaminate dataset

Identifying contaminant amplicon sequence variants (ASVs) by correlation plots: visual inspection of correlation plots of [DNA] vs relative abundance per ASV in each sample. If high negative correlation, likely contaminant (minimum 4 samples present).

N.B.: in the identification process of contaminants, data from sequence library A were excluded because the data on DNA concentration were lost.

3.1. Prepare data

```
# remove samples of 0 reads (16 samples of CS day 0)
psx <- prune_samples(sample_sums(ps1) > 0, ps1)

# extract metadata as dataframe
psx.df <- meta(psx)
psx.df$LibrarySize <- sample_sums(psx)
psx.df <- psx.df[order(psx.df$LibrarySize),] # order by library size
psx.df$Index <- seq(nrow(psx.df)) # index of samples based on rank by library size.

# file 1: samples, library size, DNA concentration
psz1 <- psx.df[, c("Description", "DNA_reading", "LibrarySize")]

# file 2: samples, taxa, reads per sample/taxon
psz2 <- as.data.frame(t(otu_table(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

# file 3: add higher taxonomic names
psx.tax <- as.data.frame(psx@tax_table)
psx.tax$OTU <- rownames(psx.tax)

# merge files
psz <- base::merge(psz, psx.tax, by = "OTU")

```

3.2. Sequencing depth per sample

```

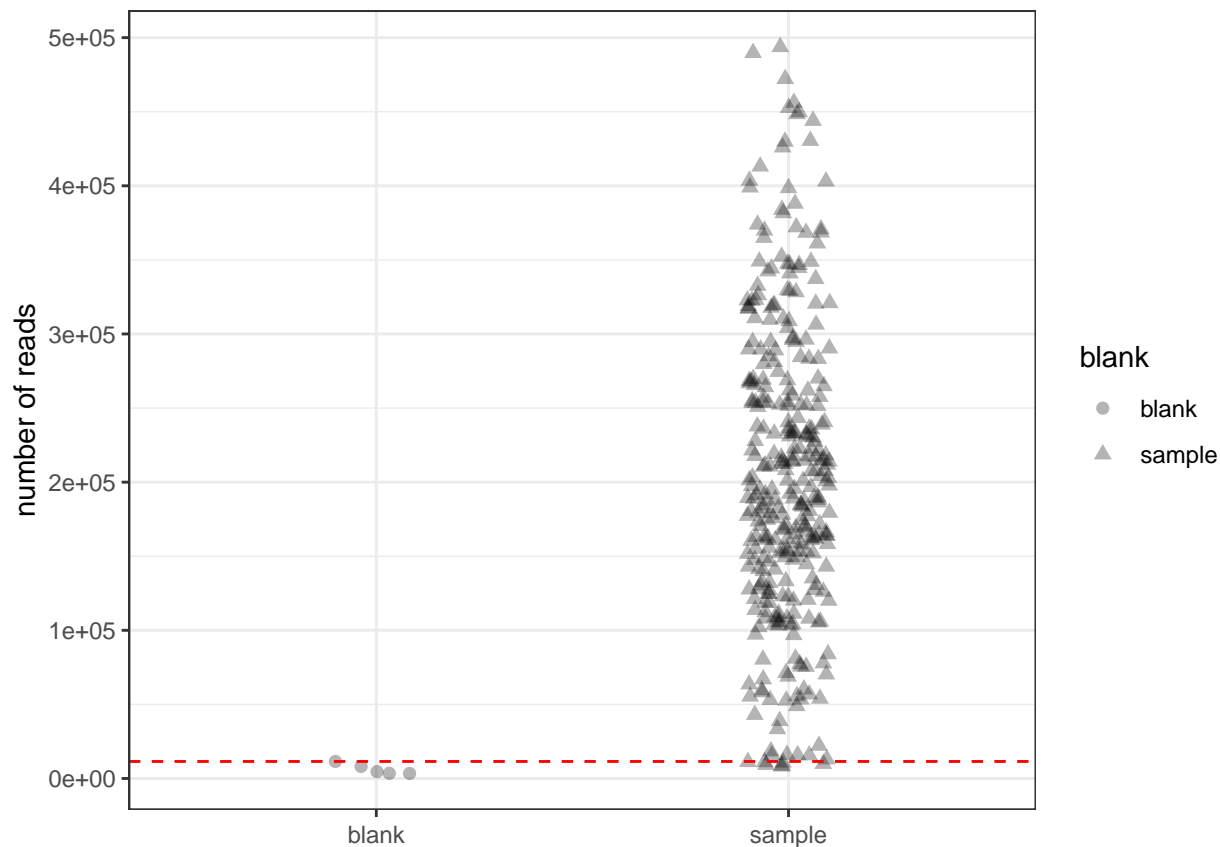
A0 <- subset(psx.df, select = c("LibrarySize", "Description"))

blanks <- c("water.A", "water.B", "water.C", "water.D", "water.E")

A0$blank <- ifelse(A0$Description %in% blanks, "blank", "sample")
maxA <- max(A0[A0$blank == "blank", 1])
View(A0)

pA0 <- ggplot(A0, aes(x = blank, y = LibrarySize)) +
  geom_jitter(aes(shape = blank), width = .1, size = 2, alpha = .3) +
  labs(x = NULL,
       y = "number of reads") +
  geom_hline(yintercept = maxA, color = "red", linetype = "dashed") +
  theme_bw() +
  theme(axis.text.x = element_text(vjust = .5, hjust = .5))
pA0

```



3.3. Correlation plots

This code creates the correlation plots that are used to identify contaminants.

N.B.: the for-loop creates a pdf with 2814 plots (1 plot per ASV), which takes a long runtime! Source: <https://stackoverflow.com/questions/26034177/save-multiple-ggplots-using-a-for-loop>

```
plot_list = list()
for (i in 1:nlevels(psz$OTU)) {
  psz.s <- psz[psz$OTU %in% levels(psz$OTU)[i],]
  p = ggplot(psz.s, aes(x = DNA_reading, y = freq)) +
    geom_point(size = 3) + labs(x = "[DNA]", y = "rel. abundance") +
    ggtitle(paste(psz.s$OTU, psz.s$Family, psz.s$Genus)) +
    geom_smooth(method = "lm") +
    scale_y_log10()
  plot_list[[i]] = p
}

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

## pdf
```

```
## 2
```

```
# result: 26 contaminant OTUs assessed visually through pdf-file inspection.  
## The following file is a handmade list based on this inspection.  
## Numbers in the file are pdf pages, corresponding to the rows in the tax_table of psx.  
visContam <- read.delim("./tables/Contaminant_OTU_by_plot.txt", header = T)
```

3.4. Plot % contaminant reads per sample

```
# add TRUE/FALSE column for contaminants in taxa table  
psx.tax2 <- psx.tax  
rownames(psx.tax2) <- NULL # reset rownames for subset based on index  
psx.tax2$contam <- rownames(psx.tax2) %in% visContam$OTU  
table(psx.tax2$contam) # works: 26 OTUs T, rest F
```

```
##  
## FALSE TRUE  
## 2788 26
```

```
rownames(psx.tax2) <- psx.tax2$OTU # restore OTU as rownames  
psx.tax2$reads <- taxa_sums(psx)
```

```
# calculate % contaminant reads  
contam.otu <- subset(psx.tax2, contam == "TRUE")  
sum(contam.otu$reads) / sum(abundances(psx))
```

```
## [1] 0.01056108
```

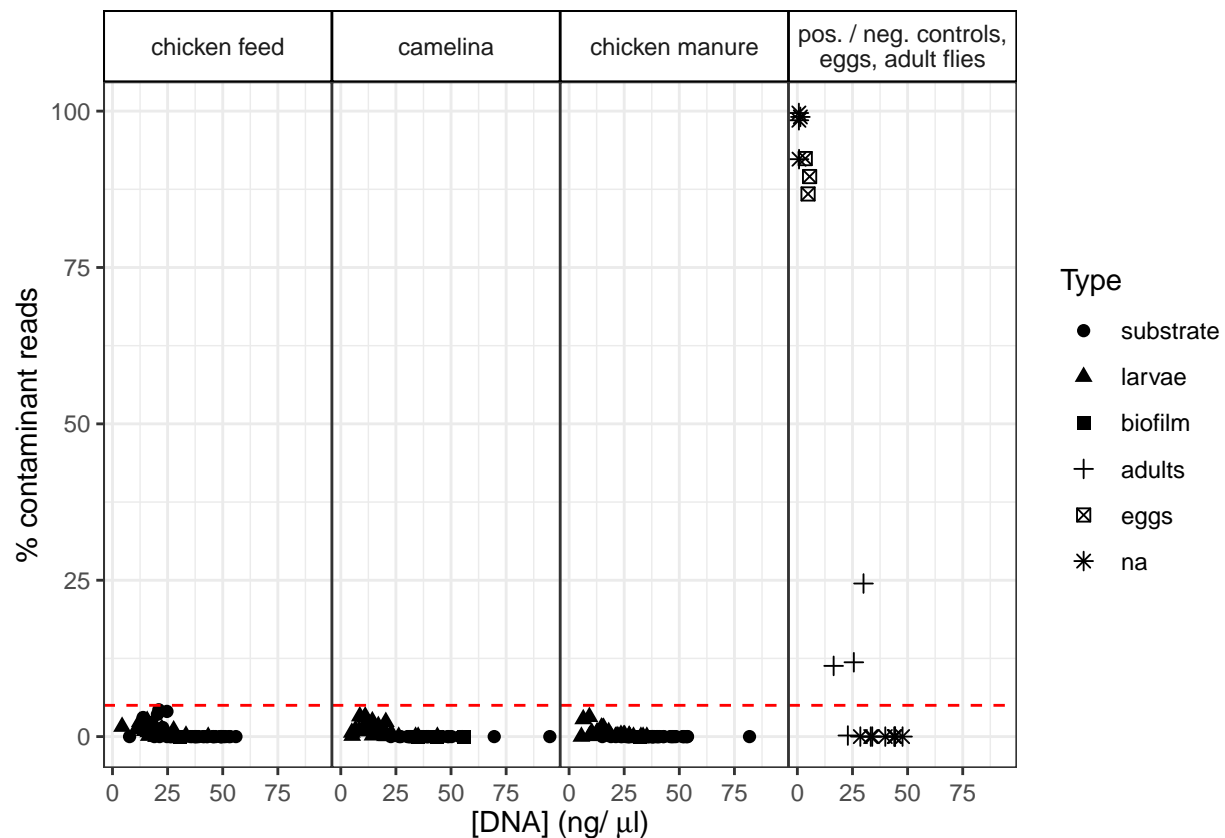
```
# 1.06% of all reads
```

```
# subset phyloseq to contaminant OTUs  
psx.sub <- subset_taxa(psx, psx.tax2$contam)
```

```
# calculate % contaminant reads per sample  
psx2.df <- meta(psx.sub)  
psx2.df$LibrarySize <- sample_sums(psx)  
psx2.df$pContam <- sample_sums(psx.sub) / sample_sums(psx)
```

```
# create plot  
pVisContam <- ggplot(psx2.df, aes(x = DNA_reading, y = 100*pContam)) +  
  geom_point(aes(shape = Type), size = 2) +  
  labs(x = expression(paste("[DNA] (ng/ ", mu, "l)", sep = "")),  
       y = "% contaminant reads") +  
  geom_hline(yintercept = 5, color = "red", linetype = "dashed") +  
  facet_grid(~Diet, labeller = as_labeller(c(  
    "CF" = "chicken feed", "CS" = "camelina",  
    "CM" = "chicken manure", "na" = "pos. / neg. controls,\neggs, adult flies")))) +  
  theme_bw() +  
  theme(panel.spacing.x = unit(0, "lines"),  
        strip.background = element_rect(colour = "black", fill = "white"),
```

```
axis.text.x = element_text(vjust = .5, hjust = .5))
pVisContam
```



Result: eggs and blanks contain many contaminant reads, adult samples up to 25%, and the experimental samples have below 5% contaminant reads (red dashed line in plot).

Percentage contaminant reads in experimental samples:

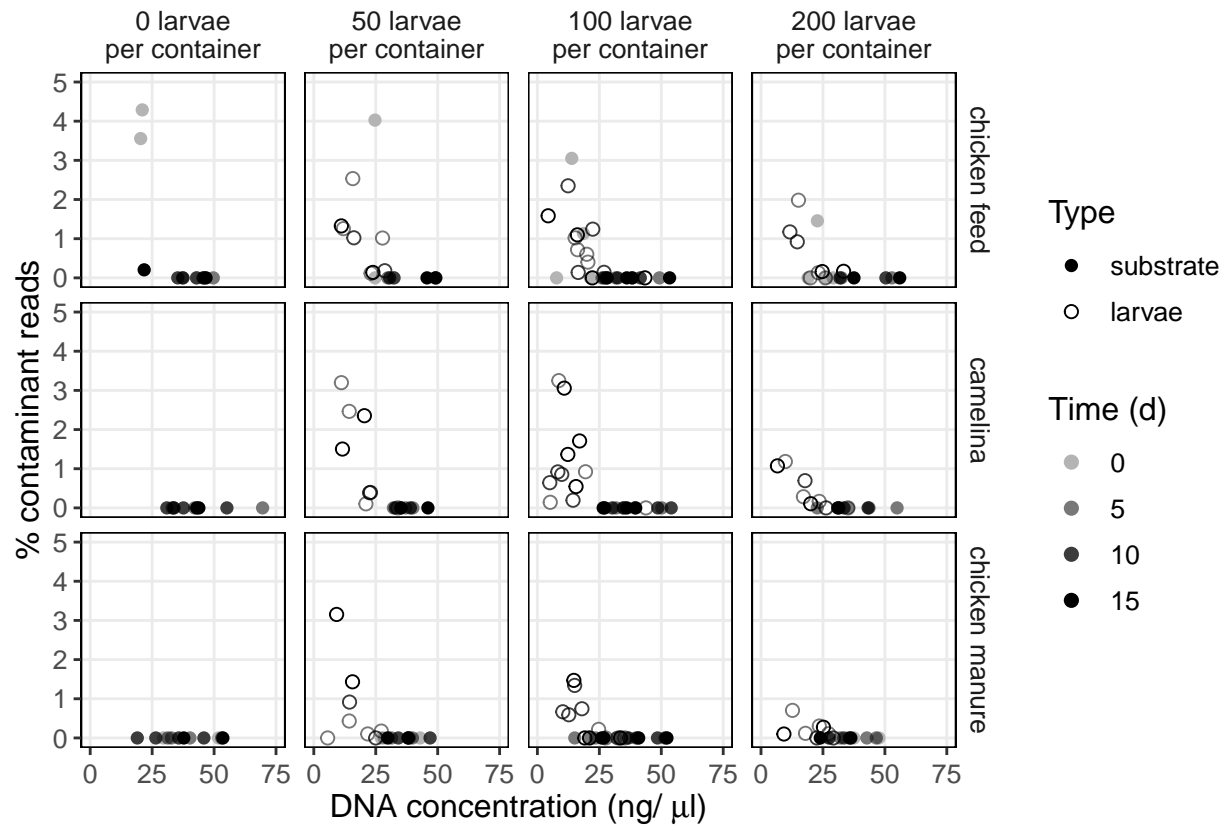
```
psx2.dfs <- subset(psx2.df, Spare == "no")
psx2.dfs$Type <- droplevels(psx2.dfs$Type)
psx2.dfs$Diet <- droplevels(psx2.dfs$Diet)
psx2.dfs$Timepoint <- droplevels(psx2.dfs$Timepoint)
psx2.dfs$Density <- droplevels(psx2.dfs$Density)

# plot
pVisContam2 <- ggplot(psx2.dfs, aes(x = DNA_reading, y = 100*pContam,
                                   alpha = Timepoint)) +
  geom_point(shape = 16, size = 2, colour = "white", alpha = 1) +
  geom_point(aes(shape = Type), size = 2) +
  scale_shape_manual(values = c(16, 1)) +
  scale_alpha_ordinal(range = c(.3, 1)) +
  labs(x = expression(paste("DNA concentration (ng/ ", mu, "l)", sep = "")),
       y = "% contaminant reads", alpha = "Time (d)") +
  scale_y_continuous(limits = c(0, 5), n.breaks = 6) +
  scale_x_continuous(limits = c(0, 75), n.breaks = 4) +
  facet_grid(Diet~Density, labeller = as_labeller(c(
    "CF" = "chicken feed", "CS" = "camelina",
```

```

"CM" = "chicken manure", "0" = "0 larvae\nper container",
"50" = "50 larvae\nper container", "100" = "100 larvae\nper container",
"200" = "200 larvae\nper container")) +
theme_bw() +
theme(panel.spacing.x = unit(.5, "lines"),
      panel.grid.minor = element_blank(),
      panel.border = element_rect(color = "black", size = .5, fill = NA),
      strip.background = element_blank(),
      axis.text.x = element_text(vjust = .5, hjust = .5),
      text = element_text(size = 12))
pVisContam2

```



3.5. Table and plot of contaminant ASVs

Supplementary Table S2 in manuscript Chapter 3 in PhD thesis and submission to *Applied and Environmental Microbiology*.

```

# table
contam.otu.clean <- contam.otu[,c(5:7,9)]
contam.otu.clean$Family_Genus <- interaction(contam.otu.clean$Family,
                                             contam.otu.clean$Genus,
                                             sep = "_", drop = T)
contam.otu.clean$Family_Genus <- gsub(pattern = "[a-z]__", replacement = "",
                                       x = contam.otu.clean$Family_Genus)

```



```

contam.otu.clean2 <- contam.otu.clean[,-c(1:2)]
rownames(contam.otu.clean2) <- NULL
contam.otu.clean2$pReads <- contam.otu.clean2$reads / sum(abundances(psx))
contam.otu.clean2 <- contam.otu.clean2[, c(
  "OTU", "Family_Genus", "reads", "pReads")]
colnames(contam.otu.clean2)[1] <- "OTU_code"
colnames(contam.otu.clean2)[3] <- "nReads"

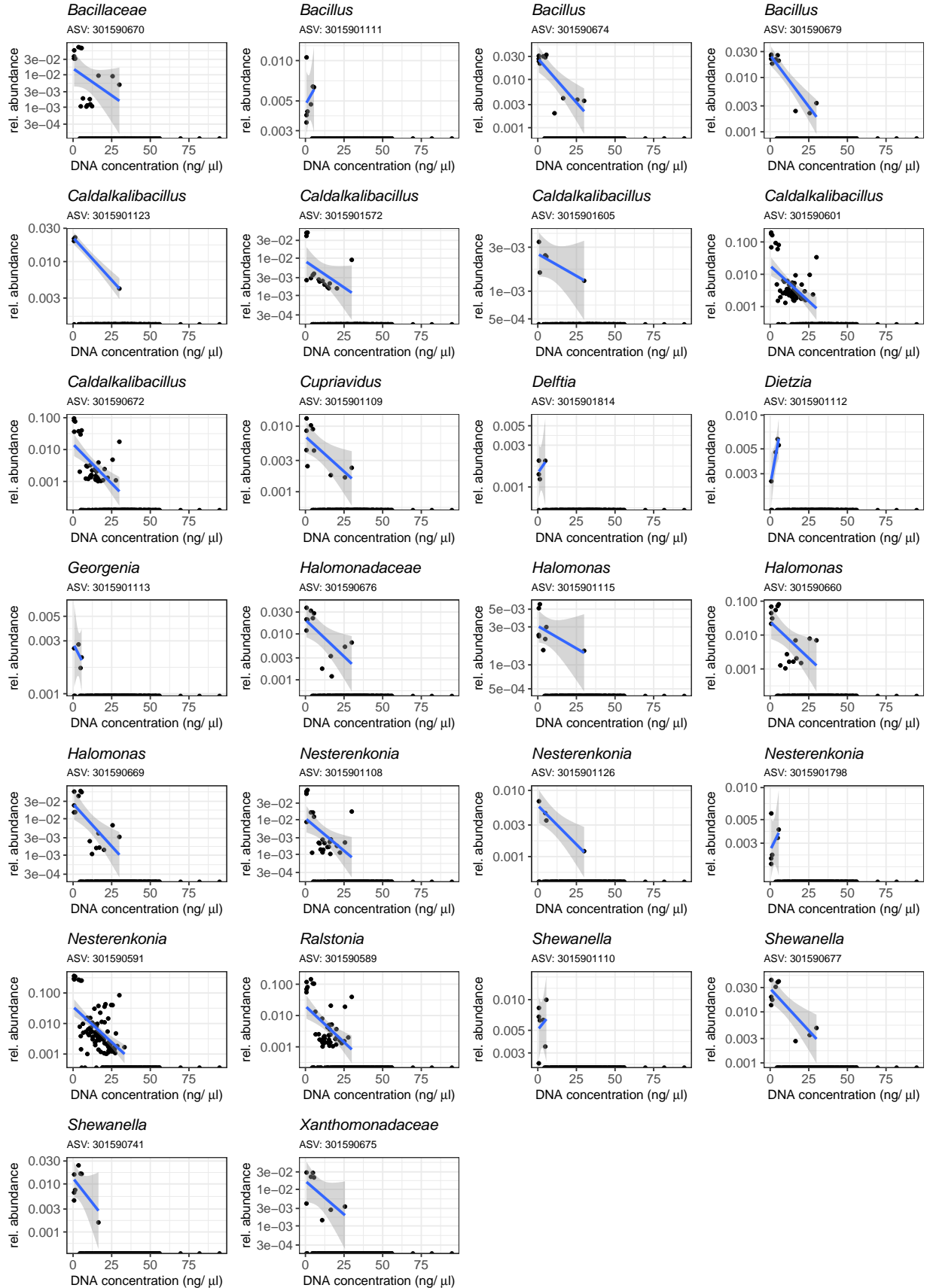
# export table
write.csv(x = contam.otu.clean2, file = "./tables/Contaminants.csv")

# plot
psz.contam <- subset(psz, OTU %in% contam.otu$OTU)
psz.contam$OTU <- droplevels(psz.contam$OTU)
psz.contam$Genus <- ifelse(psz.contam$Genus == "g__",
  yes = paste(psz.contam$Family),
  no = paste(psz.contam$Genus))
psz.contam$Genus <- sub(pattern = "[a-z]__", replacement = "", psz.contam$Genus)
psz.contam$Genus <- as.factor(psz.contam$Genus)
psz.contam <- psz.contam[order(psz.contam$Genus),]
psz.contam$OTU <- factor(psz.contam$OTU, levels(psz.contam$OTU)[unique(psz.contam$OTU)])

# for-loop
plot_list.contam = list()
for (i in 1:nlevels(psz.contam$OTU)) {
  psz.s <- psz.contam[psz.contam$OTU %in% levels(psz.contam$OTU)[i],]
  p = ggplot(psz.s, aes(x = DNA_reading, y = freq)) +
    geom_point(size = 1) +
    labs(x = expression(paste("DNA concentration (ng/ ", mu, "l)", sep = "")),
      y = "rel. abundance") +
    ggtitle(label = paste(psz.s$Genus), subtitle = paste("ASV:", psz.s$OTU)) +
    geom_smooth(method = "lm") +
    scale_y_log10() + theme_bw() +
    theme(plot.title = element_text(size = 12, face = "italic"),
      plot.subtitle = element_text(size = 8),
      axis.title = element_text(size = 10),
      axis.text = element_text(size = 10))
  plot_list.contam[[i]] = p
}

# create a combined plot with all 26 ASVs
plotContam <- ggarrange(plotlist = plot_list.contam, ncol = 4, nrow = 7,
  align = "hv", font.label = list(face = "plain"))
plotContam

```



3.6. Decontaminating the dataset

Remove the contaminant reads, identified by visual inspection of correlation plots [DNA] vs relative ASV abundance. This resulted in 26 ASVs identified as contaminants.

```
ps1.decontam <- prune_taxa(!psx.tax2$contam, ps1)
ps1.decontam <- prune_samples(sample_sums(ps1.decontam) > 0, ps1.decontam)
ps1.decontam # 2788 taxa, 334 samples

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2788 taxa and 334 samples ]
## sample_data() Sample Data: [ 334 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 2788 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2788 tips and 2787 internal nodes ]

ntaxa(ps1) - ntaxa(ps1.decontam) # 26 ASVs removed

## [1] 26

sum(abundances(ps1)) - sum(abundances(ps1.decontam)) # 715791 reads removed.

## [1] 715791
```

4. Subsetting

4.1. From total dataset ps1 (including contaminants)

```
# no-template controls (negative controls)
ps1.contr <- subset_samples(ps1, Description %in% c(
  "water.A", "water.B", "water.C", "water.D", "water.E"))
ps1.contr <- prune_taxa(taxa_sums(otu_table(ps1.contr)) > 0, ps1.contr)
ps1.contr

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 91 taxa and 5 samples ]
## sample_data() Sample Data: [ 5 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 91 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 91 tips and 90 internal nodes ]
```

4.2. From decontaminated dataset ps1.decontam

4.2.1. Quality control data

```
# positive controls (mocks)
ps1.mock <- subset_samples(ps1.decontam, Description %in% c(
  "mock3.A", "mock4.A", "mock3.B", "mock4.B", "mock3.C", "mock4.C",
  "mock3.D", "mock4.D", "mock3.E", "mock4.E"))
ps1.mock <- prune_taxa(taxa_sums(otu_table(ps1.mock)) > 0, ps1.mock)
ps1.mock
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 91 taxa and 10 samples ]
## sample_data() Sample Data: [ 10 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 91 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 91 tips and 90 internal nodes ]
```

```
# technical replicates of DNA isolation
ps1.biol <- subset_samples(ps1.decontam, Type == "substrate")
ps1.biol <- subset_samples(ps1.biol, ContainerID %in% c(1,13,17,29,33,45))
ps1.biol <- subset_samples(ps1.biol, Timepoint %in% c(5, 15))
ps1.biol <- prune_taxa(taxa_sums(otu_table(ps1.biol)) > 0, ps1.biol)
ps1.biol
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 658 taxa and 24 samples ]
## sample_data() Sample Data: [ 24 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 658 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 658 tips and 657 internal nodes ]
```

```
# technical replicates of PCR
ps1.tech <- subset_samples(ps1.decontam, Description %in% c(
  "7.C.", "6.A.", "13.G.", "7.C.T", "13.G.T", "6.A.T"))
ps1.tech <- prune_taxa(taxa_sums(otu_table(ps1.tech)) > 0, ps1.tech)
ps1.tech
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 113 taxa and 6 samples ]
## sample_data() Sample Data: [ 6 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 113 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 113 tips and 112 internal nodes ]
```

```
# biofilms and corresponding substrate samples
ps1.film <- subset_samples(ps1.decontam, Description %in% c(
  "5.H.", "17.H.", "21.H.", "24.H.", "25.H.", "41.H.", # biofilm samples
  "5.B.", "17.B.", "21.B.", "24.B.", "25.B.", "41.B.")) # substrate samples
ps1.film <- prune_taxa(taxa_sums(otu_table(ps1.film)) > 0, ps1.film)
ps1.film
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 322 taxa and 12 samples ]
## sample_data() Sample Data: [ 12 samples by 14 sample variables ]
## tax_table() Taxonomy Table: [ 322 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 322 tips and 321 internal nodes ]
```

```
# top and bottom substrate layers compared to total substrate samples
ps1.sdepth <- subset_samples(ps1.decontam, Description %in% c(
  "28.H.", "29.H.", "37.H.", "38.H.", "28.F.", "37.F."))
ps1.sdepth <- prune_taxa(taxa_sums(otu_table(ps1.sdepth)) > 0, ps1.sdepth)
ps1.sdepth
```

```
## phyloseq-class experiment-level object
```

```
## otu_table()   OTU Table:           [ 210 taxa and 6 samples ]
## sample_data() Sample Data:         [ 6 samples by 14 sample variables ]
## tax_table()   Taxonomy Table:      [ 210 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree:   [ 210 tips and 209 internal nodes ]
```

4.2.2. Experimental data

This code subsets to the experimental data, excluding:

- quality control samples (50 samples, including negative/positive controls, technical replicates, extra samples);
- low-quality samples, *i.e.* samples with fewer than 5000 reads after removal of mitochondrial, chloroplast, and contaminant reads (16 samples of camelina substrate of day 0).

```
ps1.exp <- subset_samples(ps1.decontam, Spare == "no")
ps1.exp <- prune_samples(sample_sums(otu_table(ps1.exp)) > 5000, ps1.exp)
ps1.exp <- prune_taxa(taxa_sums(otu_table(ps1.exp)) > 0, ps1.exp)
ps1.exp # 284 samples
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:           [ 2424 taxa and 284 samples ]
## sample_data() Sample Data:         [ 284 samples by 14 sample variables ]
## tax_table()   Taxonomy Table:      [ 2424 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree:   [ 2424 tips and 2423 internal nodes ]
```

5. Saving datasets in compressed RDS format

```
# raw and filtered datasets
saveRDS(psconE, "./phyobjects/psconE.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
saveRDS(ps1.film, "./phyobjects/ps1.film.rds") # biofilm samples
saveRDS(ps1.sdepth, "./phyobjects/ps1.sdepth.rds") # substrate layers

# experimental data
saveRDS(ps1.exp, "./phyobjects/ps1.exp.rds")
```