

Tests for differential abundance of taxa

Stijn Schreven

4 March 2021

Contents

Introduction	1
Load packages	2
Input files	2
1. Prepare data	2
1.1. Prepare phyloseq data at genus level	2
1.2. Subset per diet and filter to top taxa	2
1.3. Convert phyloseq object to dataframe	3
1.4. Create dataframes of paired larvae and substrate samples	4
1.5. Plot preset	4
2. Testing	4
2.1. Treatment effect, Kruskal-Wallis tests	5
2.2. Treatment effect, posthoc Wilcoxon rank-sum tests	5
2.3. Larva-substrate differences, Wilcoxon signed-rank tests	6
3. Plots: chicken manure	8
4. Export table	9

Introduction

Testing differential relative abundance of genera between:

- treatments in larval microbiota; and
- larvae and substrates per treatment.

Load packages

```
library(phyloseq)
library(microbiome)
library(microbiomeutilities)
library(purrr)
library(reshape2)
library(plyr)
library(ggplot2)
library(ggpubr)
library(viridis)
library(knitr)
```

Input files

```
pstot.g <- readRDS("./phyobjects/ps1.work.rds")
```

1. Prepare data

1.1. Prepare phyloseq data at genus level

```
pstot.g <- microbiome::aggregate_taxa(pstot.g, "Genus")
pstot.g.r <- microbiome::transform(pstot.g, "compositional")

# tax table with OTU column and best_hit
tot.tax <- as.data.frame(tax_table(pstot.g.r))
tot.tax$OTU <- rownames(tot.tax)
tax_table(pstot.g.r) <- tax_table(as.matrix(tot.tax))
pstot.g.bh <- format_to_besthit(pstot.g.r)
tot.tax.bh <- as.data.frame(tax_table(pstot.g.bh))
colnames(tot.tax.bh)[7] <- "OTU"

# remove pattern (OTU code) from $best_hit
tot.tax.bh$best_hit <- sub(pattern = "OTU-[0-9]*:", replacement = "", tot.tax.bh$best_hit)
tot.tax.bh$best_hit <- as.factor(tot.tax.bh$best_hit)

# replace the "uncultured" with [Family]:uncultured
tot.tax.bh$best_hit2 <- ifelse(tot.tax.bh$best_hit == "uncultured",
                              yes = paste(tot.tax.bh$Family, sep = " ", "uncultured"),
                              no = paste(tot.tax.bh$best_hit))
tot.tax.bh$best_hit2 <- as.factor(tot.tax.bh$best_hit2)
tot.tax.bh$best_hit2 <- revalue(tot.tax.bh$best_hit2, c("k_NA" = "unassigned taxon"))
tax_table(pstot.g.bh) <- tax_table(as.matrix(tot.tax.bh))
```

1.2. Subset per diet and filter to top taxa

Subset from main data to conserve OTU codes for genera.

```

# larvae and substrates
CF.ls <- subset_samples(pstot.g.r, Diet == "CF" & Timepoint == 15)
CF.ls <- prune_taxa(taxa_sums(otu_table(CF.ls)) > 0, CF.ls)
CM.ls <- subset_samples(pstot.g.r, Diet == "CM" & Timepoint == 15)
CM.ls <- prune_taxa(taxa_sums(otu_table(CM.ls)) > 0, CM.ls)

# chicken feed
# filter abundance threshold max > .01, prevalence > 0.1
cf1.otu <- as.data.frame(t(abundances(CF.ls)))
cf1.otu.m <- reshape2::melt(cf1.otu)
colnames(cf1.otu.m) <- c("OTU", "abund")
sum.cf.ls <- ddply(cf1.otu.m, .(OTU), summarise,
  max = max(abund),
  prev = sum(abund > 0)/length(abund))
top.cf.ls <- subset(sum.cf.ls, max > .01 & prev > .1) # 18 genera
cfls.top <- prune_taxa(taxa_names(CF.ls) %in% droplevels(top.cf.ls$OTU), CF.ls)
cfls.top <- prune_samples(sample_sums(otu_table(cfls.top)) > 0, cfls.top)

# chicken manure
# filter abundance threshold max > 0.1, prevalence > 0.1
cm1.otu <- as.data.frame(t(abundances(CM.ls)))
cm1.otu.m <- reshape2::melt(cm1.otu)
colnames(cm1.otu.m) <- c("OTU", "abund")
sum.cm.ls <- ddply(cm1.otu.m, .(OTU), summarise,
  max = max(abund),
  prev = sum(abund > 0)/length(abund))
top.cm.ls <- subset(sum.cm.ls, max > .1 & prev > .1) # 22 genera
cm1s.top <- prune_taxa(taxa_names(CM.ls) %in% droplevels(top.cm.ls$OTU), CM.ls)
cm1s.top <- prune_samples(sample_sums(otu_table(cm1s.top)) > 0, cm1s.top)

```

1.3. Convert phyloseq object to dataframe

```

# chicken feed
# file 1: metadata
cf.meta <- meta(cfls.top)
cf1 <- cf.meta[, c("Description", "ContainerID", "Diet", "Treatment", "Timepoint", "Type")]
# file 2: OTU table
cf2 <- as.data.frame(t(otu_table(cfls.top)))
cf2$Description <- rownames(cf2)
# merge files
cf3 <- merge(cf1, cf2, by = "Description")

# chicken manure
# file 1: metadata
cm.meta <- meta(cm1s.top)
cm1 <- cm.meta[, c("Description", "ContainerID", "Diet", "Treatment", "Timepoint", "Type")]
# file 2: OTU table
cm2 <- as.data.frame(t(otu_table(cm1s.top)))
cm2$Description <- rownames(cm2)
# merge files
cm3 <- merge(cm1, cm2, by = "Description")

```

```
# melt df for plotting and KW test
cf3.m <- reshape2::melt(cf3)
cm3.m <- reshape2::melt(cm3)
```

1.4. Create dataframes of paired larvae and substrate samples

The paired Wilcoxon signed-rank test requires records with paired observations, *i.e.* the relative abundances of genera in samples of larvae and substrates should be paired for each container.

```
# function to create dataframe with paired observations
paired.df <- function(df, treatment = "S/E"){
  a = subset(df, Treatment == treatment)
  a.m = reshape2::melt(a)
  a.sum = ddply(a.m, .(variable), summarise, sum = sum(value))
  a.sum0 = subset(a.sum, sum == 0)
  a.sum0$variable = droplevels(a.sum0$variable)
  a2 = subset(a.m, !variable %in% a.sum0$variable)
  a2$variable = droplevels(a2$variable)
  a2$key = interaction(a2$ContainerID, a2$variable)
  a2.s = subset(a2, Type == "substrate")
  a2.l = subset(a2, Type == "larvae")
  a3 = merge(a2.s, a2.l[,8:9], by = "key")
  return(a3)
}

# chicken feed
cf3.se <- paired.df(df = cf3, treatment = "S/E")
cf3.sie <- paired.df(df = cf3, treatment = "Si/E")
cf3.sies <- paired.df(df = cf3, treatment = "Si/Es")

# chicken manure
cm3.se <- paired.df(df = cm3, treatment = "S/E")
cm3.sie <- paired.df(df = cm3, treatment = "Si/E")
cm3.sies <- paired.df(df = cm3, treatment = "Si/Es")
cm3.sse <- paired.df(df = cm3, treatment = "Ss/E")
```

1.5. Plot preset

```
theme4 <- theme_classic() +
  theme(panel.grid.major = element_line(colour = "grey80"),
        panel.spacing = unit(.5, "lines"),
        panel.border = element_rect(color = "black", fill = NA, size = .5),
        strip.background = element_blank(),
        strip.placement = "outside",
        text = element_text(size = 20),
        axis.text.x = element_text(hjust = .5, vjust = .5))
```

2. Testing

2.1. Treatment effect, Kruskal-Wallis tests

Supplementary Table S9 in manuscript. Kruskal-Wallis tests for differences among Treatments, with FDR correction of p-values.

```
# chicken feed
cfl.kw <- data.frame("variable" = levels(cf3.m$variable), "P" = NA)
for(i in 1:nlevels(cf3.m$variable)){
  x = subset(cf3.m, Type == "larvae")
  a = levels(cf3.m$variable)[i]
  b = subset(x, variable == a)
  cfl.kw$P[i] = kruskal.test(value ~ Treatment, b)$p.value
}
cfl.kw$adjP <- p.adjust(cfl.kw$P, method = "fdr")
cfl.kw05 <- subset(cfl.kw, adjP < .05, select = c(1,3)) # 0 genera

# chicken manure
cml.kw <- data.frame("variable" = levels(cm3.m$variable), "P" = NA)
for(i in 1:nlevels(cm3.m$variable)){
  x = subset(cm3.m, Type == "larvae")
  a = levels(cm3.m$variable)[i]
  b = subset(x, variable == a)
  cml.kw$P[i] = kruskal.test(value ~ Treatment, b)$p.value
}
cml.kw$adjP <- p.adjust(cml.kw$P, method = "fdr")
cml.kw05 <- subset(cml.kw, adjP < .05, select = c(1,3)) # 17 genera.
colnames(cml.kw05)[1] <- "OTU"

# merge taxnames to KW output.
cml.kw05 <- base::merge(cml.kw05, tot.tax.bh, by = "OTU")
cml.kw05$OTU <- droplevels(cml.kw05$OTU)
```

2.2. Treatment effect, posthoc Wilcoxon rank-sum tests

Supplementary Table S8 in manuscript. Posthoc Wilcoxon rank sum tests with FDR correction of p-values. Only for chicken manure, for chicken feed there were no genera with significant differences.

```
# subset significant genera
cm4l <- subset(cm3.m, Type == "larvae" & variable %in% levels(cml.kw05$OTU))
cm4l$variable <- droplevels(cm4l$variable)

# wilcoxon tests pairwise
combn <- combn(levels(cm4l$Treatment), 2)
params <- split(as.vector(combn), rep(1:ncol(combn), each = nrow(combn)))
cml.wx <- matrix(nrow = 6, ncol = 17)
colnames(cml.wx) <- levels(cm4l$variable)
rownames(cml.wx) <- unlist(map(.x = params, .f = ~ paste0(.x, collapse = "")))

for(i in 1:nlevels(cm4l$variable)){
  a = levels(cm4l$variable)[i]
  A = subset(cm4l, variable == a)
  B = map(.x = params, .f = ~ wilcox.test(formula = value ~ Treatment,
    data = subset(A, Treatment %in% .x)))
}
```

```

C = t(data.frame(map(.x = B, .f = "p.value")))
C = as.vector(p.adjust(C, method="fdr"))
cml.wx[,i] = C
}

# subset significant genera
m.cml <- reshape2::melt(cml.wx)
m.cml <- subset(m.cml, value != "NaN") # remove NAs
s.cml <- ddply(m.cml[,2:3], .(Var2), summarise, minP = min(value))
cml.wx05 <- subset(s.cml, minP < .05)
colnames(cml.wx05)[1] <- "OTU"
cml.wx05$OTU <- as.factor(cml.wx05$OTU)
cml.wx05 <- base::merge(cml.wx05, tot.tax.bh, by = "OTU")
kable(cml.wx05[,c(1,6,7,10,2)])

```

OTU	Order	Family	best_hit2	minP
216615172	Bacillales	Bacillaceae	f__Bacillaceae	0.0083353
21661538	Bacillales	Bacillaceae	Amphibacillus	0.0055569
21661539	Bacteroidales	Dysgonomonadaceae	Proteiniphilum	0.0083353
21661549	Clostridiales	Ruminococcaceae	f__Ruminococcaceae	0.0055569
216615571	MBA03	uncultured_prokaryote	___	0.0166706
21661558	Clostridiales	Family_XI	Gottschalkia	0.0033341
21661561	Micrococcales	Micrococcaceae	Enteractinococcus	0.0083353
21661562	Bacillales	Planococcaceae	Planomicrobium	0.0166706
21661563	Sphingobacteriales	Sphingobacteriaceae	Sphingobacteriaceae	0.0166706
			uncultured	
21661574	Micrococcales	Micrococcaceae	Glutamicibacter	0.0166706
216615762	Pseudomonadales	Pseudomonadaceae	Thiopseudomonas	0.0096217
21661595	Bacillales	Bacillaceae	Pseudogracilibacillus	0.0055569

```

# vector of significant taxa
select.cm <- levels(cml.wx05$OTU) # 12 genera

```

2.3. Larva-substrate differences, Wilcoxon signed-rank tests

Wilcoxon signed-rank test for paired samples, with FDR correction of p-values. None of the FDR-corrected p-values are < 0.05.

```

# chicken feed S/E
cf.se.wx <- data.frame("variable" = levels(cf3.se$variable), "P" = NA)
for(i in 1:nlevels(cf3.se$variable)){
  a = levels(cf3.se$variable)[i]
  b = subset(cf3.se, variable == a)
  cf.se.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cf.se.wx$Padj <- p.adjust(cf.se.wx$P, method = "fdr")

# chicken feed Si/E
cf.sie.wx <- data.frame("variable" = levels(cf3.sie$variable), "P" = NA)
for(i in 1:nlevels(cf3.sie$variable)){

```

```

a = levels(cf3.sie$variable)[i]
b = subset(cf3.sie, variable == a)
cf.sie.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cf.sie.wx$Padj <- p.adjust(cf.sie.wx$P, method = "fdr")

# chicken feed Si/Es
cf.sies.wx <- data.frame("variable" = levels(cf3.sies$variable), "P" = NA)
for(i in 1:nlevels(cf3.sies$variable)){
  a = levels(cf3.sies$variable)[i]
  b = subset(cf3.sies, variable == a)
  cf.sies.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cf.sies.wx$Padj <- p.adjust(cf.sies.wx$P, method = "fdr")

# chicken manure S/E
cm.se.wx <- data.frame("variable" = levels(cm3.se$variable), "P" = NA)
for(i in 1:nlevels(cm3.se$variable)){
  a = levels(cm3.se$variable)[i]
  b = subset(cm3.se, variable == a)
  cm.se.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cm.se.wx$Padj <- p.adjust(cm.se.wx$P, method = "fdr")

# chicken manure Si/E
cm.sie.wx <- data.frame("variable" = levels(cm3.sie$variable), "P" = NA)
for(i in 1:nlevels(cm3.sie$variable)){
  a = levels(cm3.sie$variable)[i]
  b = subset(cm3.sie, variable == a)
  cm.sie.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cm.sie.wx$Padj <- p.adjust(cm.sie.wx$P, method = "fdr")

# chicken manure Si/Es
cm.sies.wx <- data.frame("variable" = levels(cm3.sies$variable), "P" = NA)
for(i in 1:nlevels(cm3.sies$variable)){
  a = levels(cm3.sies$variable)[i]
  b = subset(cm3.sies, variable == a)
  cm.sies.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cm.sies.wx$Padj <- p.adjust(cm.sies.wx$P, method = "fdr")

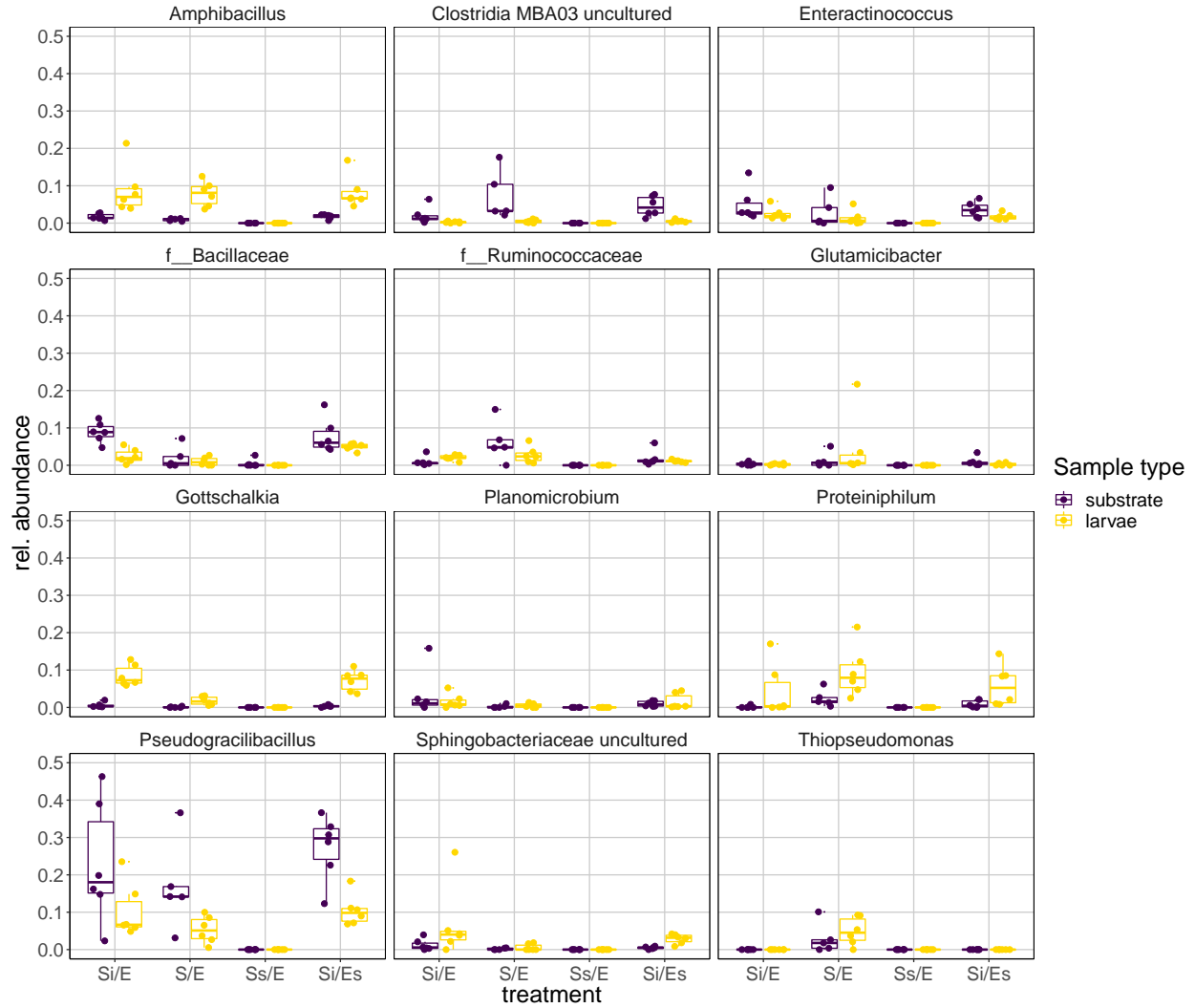
# chicken manure Ss/E
cm.sse.wx <- data.frame("variable" = levels(cm3.sse$variable), "P" = NA)
for(i in 1:nlevels(cm3.sse$variable)){
  a = levels(cm3.sse$variable)[i]
  b = subset(cm3.sse, variable == a)
  cm.sse.wx$P[i] = wilcox.test(b$value.x, b$value.y, paired = T)$p.value
}
cm.sse.wx$Padj <- p.adjust(cm.sse.wx$P, method = "fdr")

```

3. Plots: chicken manure

```
cm3.sig <- subset(cm3.m, variable %in% select.cm)
colnames(cm3.sig)[7:8] <- c("OTU", "freq")
cm3.sig2 <- merge(cm3.sig, tot.tax.bh, by = "OTU")
cm3.sig2$Treatment <- factor(cm3.sig2$Treatment, levels(cm3.sig2$Treatment)[c(2,1,4,3)])
cm3.sig2$OTU <- droplevels(cm3.sig2$OTU)
cm3.sig2$best_hit2 <- droplevels(cm3.sig2$best_hit2)
cm3.sig2$best_hit3 <- ifelse(cm3.sig2$best_hit2 == "__", "Clostridia MBA03 uncultured",
                             paste(cm3.sig2$best_hit2))

p.abund <- ggplot(cm3.sig2, aes(x = Treatment, y = freq, colour = Type)) +
  geom_boxplot(outlier.size = 0) +
  geom_point(size = 2, position = position_jitterdodge()) +
  labs(x = "treatment", y = "rel. abundance") +
  scale_y_continuous(limits = c(0, .5), n.breaks = 6) +
  scale_color_manual("Sample type", values = c("#440154FF", "gold")) +
  facet_wrap(~ best_hit3, ncol = 3, nrow = 4) + theme4
p.abund
```

```
ggsave(plot = p.abund, "./figures/Schreven_Ch4_Differences_genera_CM_treatments.png",
       w = 14, h = 12)
```

4. Export table

Supplementary Table S8 in manuscript.

```
# add omnibus test p-value to significant taxa from multiple comparisons
## between treatments of chicken manure
kw.cml.p <- subset(cml.kw05, select = 1:2, OTU %in% levels(cml.wx05$OTU))
cml.sig <- unique(base::merge(cm3.sig2[,c(1,17)], kw.cml.p, by = "OTU"))

# p-values of posthoc Wilcoxon tests
cml.wx.t <- data.frame(t(cml.wx))
colnames(cml.wx.t) <- c("S.E_Si.E", "S.E_Si.Es", "S.E_Ss.E",
                        "Si.E_Si.Es", "Si.E_Ss.E", "Si.Es_Ss.E")
cml.wx.t$OTU <- rownames(cml.wx.t)
```

```

# merge with omnibus test output
cml.sig <- base::merge(cml.sig, cml.wx.t, by = "OTU")
cml.sig[,3:9] <- round(cml.sig[,3:9], digits = 3)

# export table
write.csv(cml.sig, "./tables/Supplementary_Table_S8.csv")
# N.B.: the compact letter display used in the table in the manuscript can be obtained
## by the combined use of the boxplots in section 3 of this Markdown file and the posthoc
## Wilcoxon tests in the .CSV file.

```