

Larva-substrate dissimilarity

Stijn Schreven

4 March 2021

Contents

Introduction	1
Load packages	1
Input files	2
1. Prepare data	2
1.1. Functions	2
2. Weighted UniFrac distances	2
3. LM regression	3
3.1. Subsets	3
3.2. Chicken feed	3
3.3. Chicken manure	4
4. Errorbar plot	5

Introduction

Calculate the weighted UniFrac distance between larval and substrate microbiota of the same container, and compare these among treatments within each diet.

Load packages

```
library(phyloseq)
library(microbiome)
library(vegan)
library(plyr)
library(nlme)
library(emmeans)
```

```
library(sciplot)
library(ggplot2)
library(viridis)
```

Input files

```
tot.ls <- readRDS("./phyobjects/ps1.work.rds")
```

1. Prepare data

```
# subset to timepoint 15, for larvae and substrate samples of same container
tot.ls <- subset_samples(tot.ls, Timepoint != 0)
tot.ls <- prune_taxa(taxa_sums(otu_table(tot.ls)) > 0, tot.ls)

# genus level, relative abundance
tot.ls <- microbiome::aggregate_taxa(tot.ls, "Genus")
tot.ls <- microbiome::transform(tot.ls, "compositional")
tot.ls # 71 samples, 194 genera.
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 194 taxa and 71 samples ]
## sample_data() Sample Data: [ 71 samples by 16 sample variables ]
## tax_table() Taxonomy Table: [ 194 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 194 tips and 193 internal nodes ]
```

```
# distance matrix
wuf.ls <- distance(tot.ls, "wunifrac")
mat1 <- as.matrix(wuf.ls)
```

1.1. Functions

```
theme_ls <- theme_classic() +
  theme(text = element_text(size = 20),
    axis.text.x = element_text(vjust = .5, hjust = .5),
    strip.background = element_blank(),
    strip.placement = "outside",
    panel.grid.major = element_line(colour = "grey80"),
    panel.grid.major.x = element_blank(),
    panel.spacing = unit(.5, "lines"),
    panel.border = element_rect(color = "black", fill = NA, size = .3))
```

2. Weighted UniFrac distances

Between larval and substrate microbiota.

```

# extract distance values of larvae-substrate pairs from dist object to dataframe
#1: create dataframe
samples <- meta(tot.ls)[, c("Description", "Diet", "Type", "ContainerID",
                           "Treatment", "Timepoint")]
samples$samTime <- samples$ContainerID:samples$Timepoint
substrates <- subset(samples, Type == "substrate")
larvae <- subset(samples, Type == "larvae")
dfdist <- merge(substrates, larvae, by = "samTime")
dfdist1 <- subset(dfdist, select = -c(Type.x, Type.y, ContainerID.y,
                                     Timepoint.y, Treatment.y, Diet.y))
colnames(dfdist1) <- c("samTime", "substrate", "Diet", "ContainerID",
                      "Treatment", "Timepoint", "larvae")
dfdist1$dist.wuf <- NA

#2: fill the column "distance" with the value of each pair from matrix "mat"
for(i in 1:nrow(dfdist1)){
  a = match(dfdist1$substrate[i], labels(mat1)[[1]])
  b = match(dfdist1$larvae[i], labels(mat1)[[2]])
  dfdist1$dist.wuf[i] = mat1[a,b]
}

# summarize metrics
dist.sum <- ddply(dfdist1, ~ Diet + Treatment, summarise,
                  mean = mean(dist.wuf), sd = sd(dist.wuf), se = se(dist.wuf))

```

3. LM regression

Test per diet separately, since the design is not balanced anymore after exclusion of CF Ss/E samples.

3.1. Subsets

```

cf.dist <- dfdist1[dfdist1$Diet == "CF",]
cf.dist$Treatment <- droplevels(cf.dist$Treatment)
cm.dist <- dfdist1[dfdist1$Diet == "CM",]

```

3.2. Chicken feed

```

# variance structure selection
m.cf1 <- gls(dist.wuf ~ Treatment, method = "REML", data = cf.dist)
m.cf2 <- update(m.cf1, weights = varIdent(form = ~ 1 | Treatment))
AIC(m.cf1, m.cf2)

```

```

##          df          AIC
## m.cf1    4 -4.944874
## m.cf2    6 -1.823315

```

```
# model 1 has lowest AIC: no variance structure. Is LM.
```

```
m.cf <- lm(dist.wuf ~ Treatment, data = cf.dist)
```

```
# model output  
anova(m.cf) # n.s.
```

```
## Analysis of Variance Table  
##  
## Response: dist.wuf  
##          Df    Sum Sq   Mean Sq F value    Pr(>F)    .  
## Treatment  2 0.060881 0.0304405   3.4775 0.07604 .  
## Residuals  9 0.078782 0.0087536  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.3. Chicken manure

```
# variance structure selection  
m.cm1 <- gls(dist.wuf ~ Treatment, method = "REML", data = cm.dist)  
m.cm2 <- update(m.cm1, weights = varIdent(form = ~ 1 | Treatment))  
AIC(m.cm1, m.cm2)
```

```
##          df          AIC  
## m.cm1    5 -33.72191  
## m.cm2    8 -28.98745
```

```
# model 1 has lowest AIC, no variance structure: LM.
```

```
m.cm <- lm(dist.wuf ~ Treatment, data = cm.dist)
```

```
# model output  
anova(m.cm)
```

```
## Analysis of Variance Table  
##  
## Response: dist.wuf  
##          Df    Sum Sq   Mean Sq F value    Pr(>F)    ***  
## Treatment  3 0.214571 0.071524  17.618 1.024e-05 ***  
## Residuals 19 0.077134 0.004060  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CLD(emmeans(m.cm, ~ Treatment), Letters = letters)
```

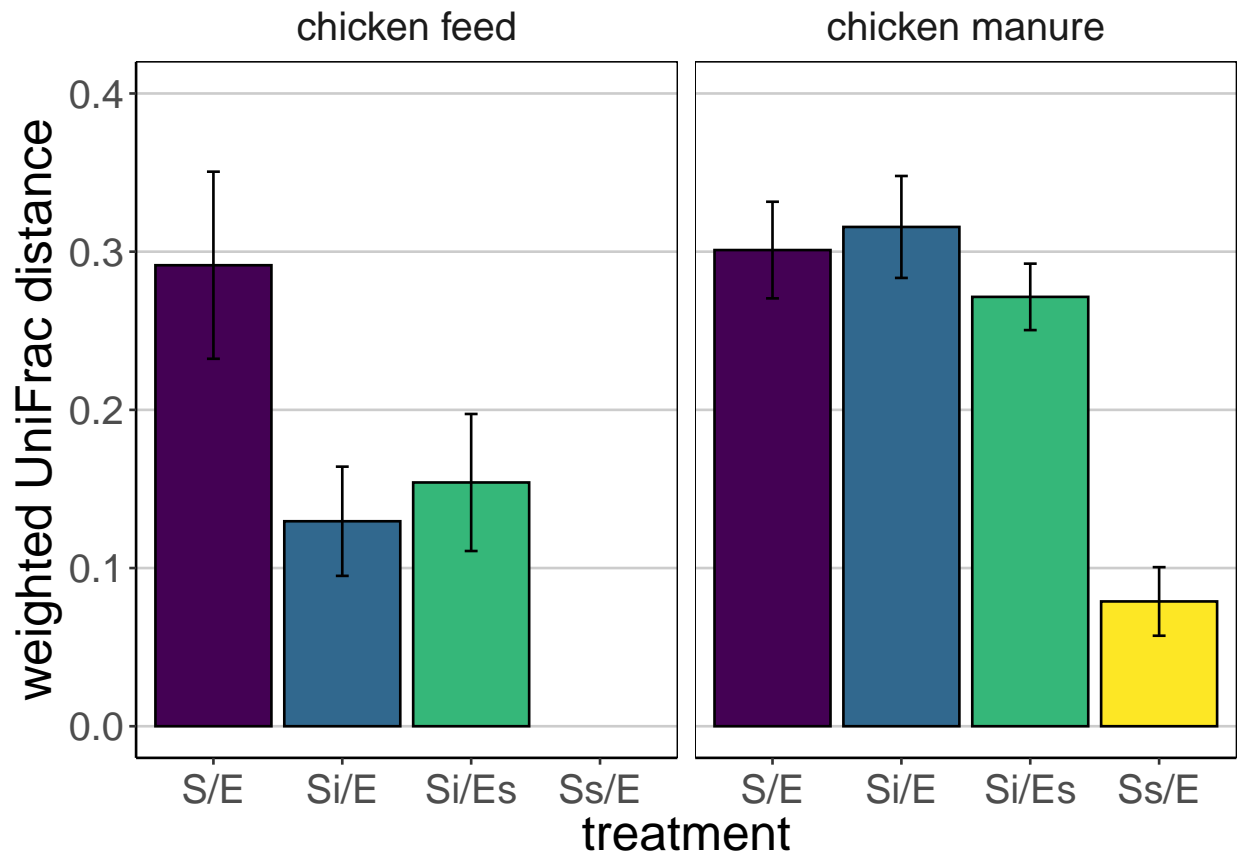
```
## Treatment emmean      SE df lower.CL upper.CL .group  
## Ss/E      0.0789 0.0260 19   0.0245   0.133   a  
## Si/Es     0.2714 0.0260 19   0.2170   0.326   b  
## S/E       0.3010 0.0285 19   0.2414   0.361   b
```

```
## Si/E      0.3156 0.0260 19   0.2612   0.370   b
##
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 4 estimates
## significance level used: alpha = 0.05
```

4. Errorbar plot

Figure 7 in manuscript. Because the models are LM, use of normal mean and SE is valid.

```
# plot
pSimwuf <- ggplot(dist.sum, aes(x = Treatment, y = mean)) +
  geom_col(aes(fill = Treatment), colour = "black") +
  scale_y_continuous(limits = c(0, .4), n.breaks = 5) +
  scale_fill_viridis(discrete = T, option = "D") +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = .1) +
  labs(y = "weighted UniFrac distance", x = "treatment") +
  facet_grid(~ Diet, labeller = as_labeller(c(CF = "chicken feed",
                                              CM = "chicken manure")))) +
  theme_ls + theme(legend.position = "none")
pSimwuf
```



```
# export plots  
ggsave(plot = pSimwuf, "./figures/Fig_7_wunifrac.png", w = 7, h = 5)  
ggsave(plot = pSimwuf, "./figures/Fig_7_wunifrac.pdf", w = 320, h = 180, u = "mm")
```