R_statistics

Vienna

6/24/2020

Summary

The experiment aimed to study the response of honey bee microbiome to three chemicals across host generations and to detect potential microbiome mediated effects on host phenotypes. ## Methods We reared adult bees under controlled lab conditions and inoculated them with a natural bee microbiome (start_pool). Bees were orally stressed with either Tetracycline, Glyphosate, Chloropthalonil or no stressor (control). Three cages per treatment were used. Both, the stress-exposed and control microbiomes were transferred to the next cycle (cage to cage transfer) which was handled as before. The third cycle aims to study effects of the pre-exposed microbiome on phenotypes of naive bee hosts in comparison to control microbiomes. For that we transferred the microbiomes and let them be established in the bee hosts without any stress factor contact. We finally applied high amounts of chemicals to bees with a control microbiome or a pre-exposed microbiome. Bee samples for 16S sequencing has been snap-frozen after cycle 1, cycle 2, cycle 3 BEFORE high stress and cycle 3 AFTER high stress. In addition, the macerated gut pools for transferring microbiome has been saved (start_microbiome as well as pool from each cage to cage transfer). We sequenced the V3-V4 region of the 16S region. Two DNA mock samples from ZymoResearch have been sequenced and named as "positive_control".

Data and metadata

Overview over all experimental variables:

Experimental variables find in metadata file:

sample_type <- microbiome_transfer or single bee *treatment* <- treatment used (control or which toxin) and additional information if microbiome transfer (e.g. Control_transfer), single bee (e.g. Control) *treatment2* <- only treatment, not indicating sample type *cage* <- cage number (three cages per treatment have been used) *treatment_cage* <- combined information of treatment and cage number (e.g. Control_2) *date* <- date of sampling during experiment *cycle* <- experimental cycle (cycle 1, cycle 2, cycle 3 before stress, cycle 3 after stress) *treatment_cycle* <- experimental cycle in combination with treatment information *treatment_cycle2*<- experimental cycle in combination with treatment plus sample type information

Statistics on survival data of bees with pre-exposed microbiomes vs respective controls under high chemical stress for main experiment

```
tetra <- read.table("R_microbiome_data_files/tetra cycle 3 day 5 to 6 survival.txt", header = TRUE)
fisher.test(tetra, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
```

```
##
## data:  tetra
## p-value = 2.573e-05
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##     3.332802 1001.464871
## sample estimates:
## odds ratio
##   23.08462
```

```
chloro <- read.table("R_microbiome_data_files/chloro cycle 3 day 5 to 6 survival.txt", header = TRUE)
fisher.test(chloro, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  chloro
## p-value = 0.0259
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1054762 0.9306751
## sample estimates:
## odds ratio
##  0.3308587
```

```
glypho <- read.table("R_microbiome_data_files/glypho cycle 3 day 5 to 7 survival.txt", header = TRUE)
fisher.test(glypho, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  glypho
## p-value = 0.8308
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.4525371 2.9079036
## sample estimates:
## odds ratio
##   1.143036
```

Glyphosate-exposed microbiomes did not significantly affect the survival of bees under high glyphosate stress, chlorothalonil-exposed microbiomes mediated protection and tetracycline-exposed microbiomes lead to higher mortality

Additional chlorothalonil experiments to figure out protective mechanisms of chlorothalonil-exposed microbiomes on bee survival. Statistics on survival data of bees with added filtered pre-exposed gut extract and added chlorothalonil vs respective controls under high chemical stress.

```
survive <- read.table("R_microbiome_data_files/Fisher_test_filtered_Chloro_control_exp.txt", header = T
fisher.test(survive, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  survive
## p-value = 0.03538
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2701351 0.9670985
## sample estimates:
## odds ratio
##  0.5135861
```

```
survive2 <- read.table("R_microbiome_data_files/Fisher_test_added_Chloro_control_exp.txt", header = TRU
fisher.test(survive2, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  survive2
## p-value = 0.6267
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2306866 2.0877885
## sample estimates:
## odds ratio
##  0.7177255
```

While filtered pre-exposed gut solution did improve later survival under high chlorothalonil stress, direct addition of chlorothalonil did not.

16S data

```
#read in otu table
otu_table=read.csv("R_microbiome_data_files/Svtab1.csv",sep=",",row.names=1)
otu_table=as.matrix(otu_table)

#read in taxonomy
taxonomy=read.csv("R_microbiome_data_files/taxonomy_modify.csv",sep=",",row.names=1)
taxonomy <- cbind(taxonomy, ASV = paste0("ASV", sprintf("%04d", 1:nrow(taxonomy))))
taxonomy=as.matrix(taxonomy)

metatable <- read.delim("R_microbiome_data_files/metadata_reorder3.txt")
#View(metatable)
row.names(metatable) <- metatable[[1]]
metatable<- metatable[,(-1)]
META <- sample_data(metatable)
```

```
phy_tree <- read_tree("R_microbiome_data_files/rooted_tree.nwk")

#import as phyloseq objects
OTU=otu_table(otu_table,taxa_are_rows=TRUE)
TAX=tax_table(taxonomy)

#create phyloseq object
ps1<- phyloseq(OTU, TAX, META, phy_tree)

# change taxonomy header
colnames(tax_table(ps1)) <- c(D0 = "Kingdom", D1 = "Phylum", D2 = "Class",
                              D3 = "Order", D4 = "Family", D5 = "Genus", D6 = "Species", ASV = "ASV")

#The total number of ASVs in the whole dataset is
length(taxa_names(ps1))
```

```
## [1] 1717
```

```
#get rid of things we do not want
ps1 = subset_taxa(ps1, Kingdom =="Bacteria")
ps1 <- prune_taxa(taxa_sums(ps1) > 0, ps1)
ps1<-subset_taxa(ps1, (Order!="Chloroplast"))
ps1<-subset_taxa(ps1, (Family!="Mitochondria"))
length(taxa_names(ps1))
```

```
## [1] 1167
```

```
#reduces total taxa numbers from 1717 to 1167 (minus 550)

# mean, max and min of sample read counts
smin <- min(sample_sums(ps1))
smean <- mean(sample_sums(ps1))
smax <- max(sample_sums(ps1))
# printing the results
cat("The minimum sample read count is:",smin)
```

```
## The minimum sample read count is: 12351
```

```
cat("The average sample read count is:",smean)
```

```
## The average sample read count is: 29843.03
```

```
cat("The maximum sample read count is:",smax)
```

```
## The maximum sample read count is: 66542
```

comparing difference between methods

We sequenced to types of samples: whole bee abdomen and the mix of three macerated guts for cage to cage transfers after each cycle Therefore, we need to test if these samples are different due to the differences in the methods prior extracting before deciding if we include all or not. Use PERMANOVA on bray-curtis dissimilarities using proportional transformed abundance data

```r
set.seed(42)
methods = ps1
methods <- transform_sample_counts(methods, function(OTU) {OTU / sum(OTU)})

Control = subset_samples(methods, treatment2 == "Control")
Control <- prune_taxa(taxa_sums(Control) > 0, Control)
C_metadata <- as(sample_data(Control), "data.frame")
adonis(distance(Control, method="bray") ~ sample_type,data = C_metadata,perm=999)
```

```
##
## Call:
## adonis(formula = distance(Control, method = "bray") ~ sample_type,      data = C_metadata, permutatio
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## sample_type  1   0.05335 0.053349 0.74896 0.03292  0.516
## Residuals   22   1.56707 0.071231         0.96708
## Total       23   1.62042                  1.00000
```

```r
Tetra = subset_samples(methods, treatment2 == "Tetracycline")
Tetra <- prune_taxa(taxa_sums(Tetra) > 0, Tetra)
T_metadata <- as(sample_data(Tetra), "data.frame")
adonis(distance(Tetra, method="bray") ~ sample_type,data = T_metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(Tetra, method = "bray") ~ sample_type,      data = T_metadata, permutations
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## sample_type  1   0.05033 0.050334 0.71685 0.03023  0.549
## Residuals   23   1.61498 0.070216         0.96977
## Total       24   1.66531                  1.00000
```

```r
Glypho = subset_samples(methods, treatment2 == "Glyphosate")
Glypho <- prune_taxa(taxa_sums(Glypho) > 0, Glypho)
G_metadata <- as(sample_data(Glypho), "data.frame")
adonis(distance(Glypho, method="bray") ~ sample_type,data = G_metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(Glypho, method = "bray") ~ sample_type,      data = G_metadata, permutation
##
```

```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs  MeanSqs F.Model   R2 Pr(>F)
## sample_type  1    0.0709 0.070933 0.79591 0.02  0.541
## Residuals   39    3.4757 0.089122         0.98
## Total       40    3.5467                  1.00
```

```
Chloro = subset_samples(methods, treatment2 == "Chlorothalonil")
Chloro <- prune_taxa(taxa_sums(Chloro) > 0, Chloro)
Ch_metadata <- as(sample_data(Chloro), "data.frame")
adonis(distance(Chloro, method="bray") ~ sample_type,data = Ch_metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(Chloro, method = "bray") ~ sample_type,      data = Ch_metadata, permutati
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## sample_type  1    0.0715 0.071501  1.2362 0.02998  0.253
## Residuals   40    2.3137 0.057842         0.97002
## Total       41    2.3852                  1.00000
```

No significant difference between gut pools or whole bees in any treatment.

Alpha diversity statistics

make statistical comparisons on Observed species numbers - comparing treatments against respective controls in each cycle

```
alpha<- ps1
alpha1 = subset_samples(alpha, cycle=="cycle_one" | cycle=="cycle_two" | cycle == "cycle_three_before_s
#rarefy to even numbers
set.seed(1)
alpha1_ra <- rarefy_even_depth(alpha1,sample.size=12351, replace=FALSE, rngseed = 1)
```

```
## `set.seed(1)` was used to initialize repeatable random subsampling.

## Please record this for your records so others can reproduce.

## Try `set.seed(1); .Random.seed` for the full vector

## ...
## 435OTUs were removed because they are no longer
## present in any sample after random subsampling

## ...
```

```r
results = estimate_richness(alpha1_ra, measures = 'Observed')
d = sample_data(alpha1_ra)

# calculate wilcox-test
Control_1 = results[d[,'treatment_cycle3'] == 'Control_cycle_1',]
Tetracycline_1 = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_1',]
Glyphosate_1 = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_1',]
Chlorothalonil_1 = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_1',]
wilcox.test(Control_1, Chlorothalonil_1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_1 and Chlorothalonil_1
## W = 98, p-value = 0.1391
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Control_1, Glyphosate_1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_1 and Glyphosate_1
## W = 116, p-value = 0.01184
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Control_1, Tetracycline_1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_1 and Tetracycline_1
## W = 36, p-value = 0.0111
## alternative hypothesis: true location shift is not equal to 0
```

```r
pvalues<-c(0.139,0.0184,0.0111)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.1390 0.0276 0.0276
```

```r
#cycle 2
Control_2 = results[d[,'treatment_cycle3'] == 'Control_cycle_2',]
Tetracycline_2 = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_2',]
Glyphosate_2 = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_2',]
Chlorothalonil_2 = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_2',]
wilcox.test(Control_2, Chlorothalonil_2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_2 and Chlorothalonil_2
## W = 50.5, p-value = 0.2218
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Control_2, Glyphosate_2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_2 and Glyphosate_2
## W = 78, p-value = 0.4769
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Control_2, Tetracycline_2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Control_2 and Tetracycline_2
## W = 144, p-value = 3.449e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
pvalues<-c(0.222,0.477,3.449e-05)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.33300000 0.47700000 0.00010347
```

```
#cycle 3
control_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'control_cycle_3_before_stress',]
Tetracycline_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_3_before_stre
Glyphosate_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_3_before_stress'
Chlorothalonil_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_3_before_
wilcox.test(control_cycle_3_before_stress, Chlorothalonil_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  control_cycle_3_before_stress and Chlorothalonil_cycle_3_before_stress
## W = 72, p-value = 0.07223
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(control_cycle_3_before_stress, Glyphosate_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  control_cycle_3_before_stress and Glyphosate_cycle_3_before_stress
## W = 178.5, p-value = 0.03822
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(control_cycle_3_before_stress, Tetracycline_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  control_cycle_3_before_stress and Tetracycline_cycle_3_before_stress
## W = 243, p-value = 9.364e-06
## alternative hypothesis: true location shift is not equal to 0
```

```r
pvalues<-c(0.072,0.038,9.364e-06)
p.adjust(pvalues,method="fdr")
```

```
## [1] 7.2000e-02 5.7000e-02 2.8092e-05
```

Numbers of observed species is significantly different under tetracycline to the control in all three cycles, while glyphosate affected the observed species number significantly in cycle 1 and chlorothalonil had no significant effect at any time point.

make statistical comparisons on Shannon alpha diversity index - comparing treatments against respective controls in each cycle

```r
alpha<- ps1
alpha1 = subset_samples(alpha, cycle=="cycle_one" | cycle=="cycle_two" | cycle == "cycle_three_before_s
#rarefy to even numbers
set.seed(1)
alpha1_ra <- rarefy_even_depth(alpha1,sample.size=12351, replace=FALSE, rngseed = 1)
```

```
## `set.seed(1)` was used to initialize repeatable random subsampling.

## Please record this for your records so others can reproduce.

## Try `set.seed(1); .Random.seed` for the full vector

## ...

## 435OTUs were removed because they are no longer
## present in any sample after random subsampling

## ...
```

```r
results = estimate_richness(alpha1_ra, measures = 'Shannon')
d = sample_data(alpha1_ra)

# calculate wilcox-test
Control_1 = results[d[,'treatment_cycle3'] == 'Control_cycle_1',]
Tetracycline_1 = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_1',]
Glyphosate_1 = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_1',]
Chlorothalonil_1 = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_1',]
wilcox.test(Control_1, Chlorothalonil_1)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_1 and Chlorothalonil_1
## W = 105, p-value = 0.05966
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Control_1, Glyphosate_1)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_1 and Glyphosate_1
## W = 122, p-value = 0.002914
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Control_1, Tetracycline_1)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_1 and Tetracycline_1
## W = 36, p-value = 0.004396
## alternative hypothesis: true location shift is not equal to 0
```

```r
pvalues<-c(0.0596,0.002914,0.004396)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.059600 0.006594 0.006594
```

```r
#cycle 2
Control_2 = results[d[,'treatment_cycle3'] == 'Control_cycle_2',]
Tetracycline_2 = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_2',]
Glyphosate_2 = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_2',]
Chlorothalonil_2 = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_2',]
wilcox.test(Control_2, Chlorothalonil_2)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_2 and Chlorothalonil_2
## W = 64, p-value = 0.6707
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Control_2, Glyphosate_2)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_2 and Glyphosate_2
## W = 64, p-value = 0.9279
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Control_2, Tetracycline_2)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Control_2 and Tetracycline_2
## W = 144, p-value = 7.396e-07
## alternative hypothesis: true location shift is not equal to 0
```

```
pvalues<-c(0.671,0.9279,7.396e-07)
p.adjust(pvalues,method="fdr")
```

```
## [1] 9.2790e-01 9.2790e-01 2.2188e-06
```

```
#cycle 3
control_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'control_cycle_3_before_stress',]
Tetracycline_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Tetracycline_cycle_3_before_str
Glyphosate_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Glyphosate_cycle_3_before_stress'
Chlorothalonil_cycle_3_before_stress = results[d[,'treatment_cycle3'] == 'Chlorothalonil_cycle_3_before_
wilcox.test(control_cycle_3_before_stress, Chlorothalonil_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test
##
## data:  control_cycle_3_before_stress and Chlorothalonil_cycle_3_before_stress
## W = 108, p-value = 0.6406
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(control_cycle_3_before_stress, Glyphosate_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test
##
## data:  control_cycle_3_before_stress and Glyphosate_cycle_3_before_stress
## W = 213, p-value = 0.0003847
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(control_cycle_3_before_stress, Tetracycline_cycle_3_before_stress)
```

```
##
##  Wilcoxon rank sum test
##
## data:  control_cycle_3_before_stress and Tetracycline_cycle_3_before_stress
## W = 243, p-value = 2.124e-08
## alternative hypothesis: true location shift is not equal to 0
```

```
pvalues<-c(0.641,0.000385,2.124e-08)
p.adjust(pvalues,method="fdr")
```

```
## [1] 6.410e-01 5.775e-04 6.372e-08
```

Under tetracycline the Shannon alpha diversity was significantly affected in all three cycles. Glyphosate significantly affected the alpha diversity in cycle 1 and 3 and chlorothalonil did not had any significant effect.

Beta diversity

test difference between treatments and respective controls in the three cycles to statistically verify microbial community compositional difference seen in ordination plots

```
set.seed(42)
compare=ps1
compare1 <- transform_sample_counts(compare, function(OTU) {OTU / sum(OTU)})

cycle1 = subset_samples(compare1, cycle == "cycle_one")
cycle1 <- prune_taxa(taxa_sums(cycle1) > 0, cycle1)
cycle1.1 <- as(sample_data(cycle1), "data.frame")

cycle2 = subset_samples(compare1, cycle == "cycle_two")
cycle2 <- prune_taxa(taxa_sums(cycle2) > 0, cycle2)
cycle2.1 <- as(sample_data(cycle2), "data.frame")

cycle3b = subset_samples(compare1, cycle == "cycle_three_before_stress")
cycle3b <- prune_taxa(taxa_sums(cycle3b) > 0, cycle3b)
cycle3b.1 <- as(sample_data(cycle3b), "data.frame")

#cycle 1

subs <- subset_samples(cycle1, treatment2 %in% c("Control", "Chlorothalonil"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations =
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment2  1   0.07746 0.077456  1.3139 0.05636  0.211
## Residuals  22   1.29693 0.058951         0.94364
## Total      23   1.37439                  1.00000
```

```
subs <- subset_samples(cycle1, treatment2 %in% c("Control", "Glyphosate"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2, data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations =
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment2  1   0.23951 0.239510  3.1008 0.12353  0.022 *
## Residuals  22   1.69932 0.077242         0.87647
## Total      23   1.93883                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs <- subset_samples(cycle1, treatment2 %in% c("Control", "Tetracycline"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations = 
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## treatment2  1   1.39916 1.39916  31.186 0.70579  0.004 **
## Residuals  13   0.58325 0.04487         0.29421
## Total      14   1.98241                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pvalues <- c(0.21,0.022,0.004)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.210 0.033 0.012
```

```
#cycle 2
subs <- subset_samples(cycle2, treatment2 %in% c("Control", "Chlorothalonil"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations = 
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
```

```
##             Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment2  1   0.04865 0.048654  0.7824 0.03434  0.492
## Residuals  22   1.36809 0.062186         0.96566
## Total      23   1.41675                  1.00000
```

```
subs <- subset_samples(cycle2, treatment2 %in% c("Control", "Glyphosate"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations =
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment2  1   0.14948 0.149477  1.9993 0.08693  0.079 .
## Residuals  21   1.57005 0.074764         0.91307
## Total      22   1.71952                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs <- subset_samples(cycle2, treatment2 %in% c("Control", "Tetracycline"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment2,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment2,      data = metadata, permutations =
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## treatment2  1    3.6437  3.6437  62.591 0.73992  0.001 ***
## Residuals  22    1.2807  0.0582         0.26008
## Total      23    4.9244                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pvalues <- c(0.5,0.08,0.001)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.500 0.120 0.003
```

```
#cycle 3 before stress

subs <- subset_samples(cycle3b, treatment3 %in% c("Control", "Chlorothalonil"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment3,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment3,      data = metadata, permutations = 9
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment3  1   0.07295 0.072949  1.5559 0.04376  0.158
## Residuals  34   1.59406 0.046884         0.95624
## Total      35   1.66700                  1.00000
```

```
subs <- subset_samples(cycle3b, treatment3 %in% c("Control", "Glyphosate"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment3,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment3,      data = metadata, permutations = 9
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment3  1   0.25456 0.254564  4.9932 0.12805  0.002 **
## Residuals  34   1.73338 0.050982         0.87195
## Total      35   1.98795                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs <- subset_samples(cycle3b, treatment3 %in% c("Control", "Tetracycline"))
metadata <- as(sample_data(subs), "data.frame")
adonis(distance(subs, method="bray") ~ treatment3,data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs, method = "bray") ~ treatment3,      data = metadata, permutations = 9
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
```

```
##
##            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## treatment3  1    2.5924 2.59238  46.491 0.57759  0.001 ***
## Residuals  34    1.8959 0.05576         0.42241
## Total      35    4.4882                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pvalues <- c(0.158,0.002,0.001)
p.adjust(pvalues,method="fdr")
```

```
## [1] 0.158 0.003 0.003
```

Community composition was significantly affected under tetracycline in all cycles. Glyphosate affected it in cycle 1 and 3 while chlorothalonil did not show to cause significant changes

Test general effects of treatment across cycles

```
set.seed(42)
compare=ps1
compare <- prune_taxa(taxa_sums(compare) > 0, compare)
compare <- transform_sample_counts(compare, function(OTU) {OTU / sum(OTU)})
compare.1 <- as(sample_data(compare), "data.frame")

cycle1 = subset_samples(compare, cycle == "cycle_one")
cycle1 <- prune_taxa(taxa_sums(cycle1) > 0, cycle1)
cycle1.2 <- as(sample_data(cycle1), "data.frame")

cycle2 = subset_samples(compare, cycle == "cycle_two")
cycle2 <- prune_taxa(taxa_sums(cycle2) > 0, cycle2)
cycle2.2 <- as(sample_data(cycle2), "data.frame")

cycle3_before_stress = subset_samples(compare, cycle == "cycle_three_before_stress")
cycle3_before_stress <- prune_taxa(taxa_sums(cycle3_before_stress) > 0, cycle3_before_stress)
cycle3_before_stress.2 <- as(sample_data(cycle3_before_stress), "data.frame")

d = distance(compare, "bray")
adonis(d ~ treatment3, compare.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ treatment3, data = compare.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##             Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## treatment3   9    15.696 1.74397   24.73 0.55014  0.001 ***
## Residuals  182    12.835 0.07052         0.44986
## Total      191    28.531                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
d1 = distance(cycle1, "bray")
adonis(d1 ~ treatment3, cycle1.2, perm=999)
```

```
##
## Call:
## adonis(formula = d1 ~ treatment3, data = cycle1.2, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
## treatment3  3    1.8842 0.62807  9.0261 0.4362  0.001 ***
## Residuals  35    2.4354 0.06958         0.5638
## Total      38    4.3196                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
d2 = distance(cycle2, "bray")
adonis(d2 ~ treatment3, cycle2.2, perm=999)
```

```
##
## Call:
## adonis(formula = d2 ~ treatment3, data = cycle2.2, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
## treatment3  3    5.6894 1.89647  30.557 0.6807  0.001 ***
## Residuals  43    2.6687 0.06206         0.3193
## Total      46    8.3581                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
d3 = distance(cycle3_before_stress, "bray")
adonis(d3 ~ treatment3, cycle3_before_stress.2, perm=999)
```

```
##
## Call:
## adonis(formula = d3 ~ treatment3, data = cycle3_before_stress.2,      permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
```

```
##
##            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## treatment3  3    3.3069 1.10229  22.015 0.56913  0.001 ***
## Residuals  50    2.5035 0.05007         0.43087
## Total      53    5.8104                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Treatment significantly explains differences in microbiome in all cycles (between 43 and 65 % of variation).

Test for cage effects

```
set.seed(42)
cycle1_chloro = subset_samples(cycle1, treatment3 == "Chlorothalonil")
cycle1_chloro.1 <- as(sample_data(cycle1_chloro), "data.frame")
d = distance(cycle1_chloro, "bray")
adonis(d ~ cage, cycle1_chloro.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle1_chloro.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model     R2 Pr(>F)
## cage       2   0.14049 0.070243  1.0818 0.1938  0.369
## Residuals  9   0.58440 0.064934         0.8062
## Total     11   0.72489                  1.0000
```

```
cycle1_gl = subset_samples(cycle1, treatment3 == "Glyphosate")
cycle1_gl.1 <- as(sample_data(cycle1_gl), "data.frame")
d = distance(cycle1_gl, "bray")
adonis(d ~ cage, cycle1_gl.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle1_gl.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.13139 0.065696 0.59371 0.11656   0.75
## Residuals  9   0.99589 0.110655         0.88344
## Total     11   1.12728                  1.00000
```

```
cycle1_co = subset_samples(cycle1, treatment3 == "Control")
cycle1_co.1 <- as(sample_data(cycle1_co), "data.frame")
d = distance(cycle1_co, "bray")
adonis(d ~ cage, cycle1_co.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle1_co.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.09317 0.046587 0.87558 0.16288  0.523
## Residuals  9   0.47887 0.053207         0.83712
## Total     11   0.57204                  1.00000
```

```
#cycle2
cycle2_control = subset_samples(cycle2, treatment3 == "Control")
cycle2_control.1 <- as(sample_data(cycle2_control), "data.frame")
d = distance(cycle2_control, "bray")
adonis(d ~ cage, cycle2_control.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle2_control.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.36945 0.184726  4.0989 0.47668  0.001 ***
## Residuals  9   0.40561 0.045067         0.52332
## Total     11   0.77506                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cycle2_chloro = subset_samples(cycle2, treatment3 == "Chlorothalonil")
cycle2_chloro.1 <- as(sample_data(cycle2_chloro), "data.frame")
d = distance(cycle2_chloro, "bray")
adonis(d ~ cage, cycle2_chloro.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle2_chloro.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
```

```
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.11535 0.057676  1.0867 0.19451  0.338
## Residuals  9   0.47768 0.053076         0.80549
## Total     11   0.59304                  1.00000
```

```
cycle2_gl = subset_samples(cycle2, treatment3 == "Glyphosate")
cycle2_gl.1 <- as(sample_data(cycle2_gl), "data.frame")
d = distance(cycle2_gl, "bray")
adonis(d ~ cage, cycle2_gl.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle2_gl.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.27728 0.138641  2.1424 0.34879  0.058 .
## Residuals  8   0.51771 0.064713         0.65121
## Total     10   0.79499                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cycle2_tet = subset_samples(cycle2, treatment3 == "Tetracycline")
cycle2_tet.1 <- as(sample_data(cycle2_tet), "data.frame")
d = distance(cycle2_tet, "bray")
adonis(d ~ cage, cycle2_tet.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle2_tet.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.07478 0.037391 0.78103 0.14789  0.511
## Residuals  9   0.43086 0.047874         0.85211
## Total     11   0.50564                  1.00000
```

```
#cycle 3 before stress
cycle3_control = subset_samples(cycle3_before_stress, treatment3 == "Control")
cycle3_control.1 <- as(sample_data(cycle3_control), "data.frame")
d = distance(cycle3_control, "bray")
adonis(d ~ cage, cycle3_control.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle3_control.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.15617 0.078084  1.5568 0.11484  0.168
## Residuals 24   1.20372 0.050155         0.88516
## Total     26   1.35989                  1.00000
```

```r
cycle3_chloro = subset_samples(cycle3_before_stress, treatment3 == "Chlorothalonil")
cycle3_chloro.1 <- as(sample_data(cycle3_chloro), "data.frame")
d = distance(cycle3_chloro, "bray")
adonis(d ~ cage, cycle3_chloro.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle3_chloro.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.06693 0.033465  1.2006 0.28582  0.286
## Residuals  6   0.16723 0.027872         0.71418
## Total      8   0.23417                  1.00000
```

```r
cycle3_gl = subset_samples(cycle3_before_stress, treatment3 == "Glyphosate")
cycle3_gl.1 <- as(sample_data(cycle3_gl), "data.frame")
d = distance(cycle3_gl, "bray")
adonis(d ~ cage, cycle3_gl.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle3_gl.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.22242 0.111209  4.4167 0.59551  0.008 **
## Residuals  6   0.15108 0.025179         0.40449
## Total      8   0.37349                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cycle3_tet = subset_samples(cycle3_before_stress, treatment3 == "Tetracycline")
cycle3_tet.1 <- as(sample_data(cycle3_tet), "data.frame")
d = distance(cycle3_tet, "bray")
adonis(d ~ cage, cycle3_tet.1, perm=999)
```

```
##
## Call:
## adonis(formula = d ~ cage, data = cycle3_tet.1, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cage       2   0.11974 0.059872 0.86309 0.22342  0.569
## Residuals  6   0.41622 0.069370         0.77658
## Total      8   0.53596                  1.00000
```

Testing if the three cages within a treatment show significant variations in microbiome composition (cage effects) reveals that a cage effect is not common in our data. Only in cycle 2 control and cycle 3 glyphosate significant cage variations are seen.

Time effect

Test if microbial composition of treatments differ across the three cycles

```
set.seed(42)
compare = ps1
compare2 <- prune_taxa(taxa_sums(compare) > 0, compare)
compare2 <- transform_sample_counts(compare2, function(OTU) {OTU / sum(OTU)})

subs1 <- subset_samples(compare2, treatment_cycle3 == "Control_cycle_1"|treatment_cycle3=="Control_cycle
metadata <- as(sample_data(subs1), "data.frame")
adonis(distance(subs1, method="bray") ~ cycle,
       data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs1, method = "bray") ~ cycle, data = metadata,      permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cycle      2    0.6091 0.304525  5.3998 0.18367  0.002 **
## Residuals 48    2.7070 0.056396         0.81633
## Total     50    3.3160                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs2 <- subset_samples(compare2, treatment_cycle =="Chlorothalonil_cycle_1"|treatment_cycle== "Chlorot
metadata <- as(sample_data(subs2), "data.frame")
adonis(distance(subs2, method="bray") ~ cycle,
       data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs2, method = "bray") ~ cycle, data = metadata,      permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cycle      2   0.46568 0.232842  4.5006 0.23079  0.002 **
## Residuals 30   1.55209 0.051736         0.76921
## Total     32   2.01778                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs3 <- subset_samples(compare2, treatment_cycle3 %in% c("Glyphosate_cycle_1", "Glyphosate_cycle_2","Gl
metadata <- as(sample_data(subs3), "data.frame")
adonis(distance(subs3, method="bray") ~ cycle,
       data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs3, method = "bray") ~ cycle, data = metadata,      permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cycle      2   0.75172 0.37586  4.7478 0.24667  0.001 ***
## Residuals 29   2.29577 0.07916         0.75333
## Total     31   3.04748                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
subs4 <- subset_samples(compare2, treatment_cycle %in% c("Tetracycline_cycle_1", "Tetracycline_cycle_2"
metadata <- as(sample_data(subs4), "data.frame")
adonis(distance(subs4, method="bray") ~ cycle,
       data = metadata, perm=999)
```

```
##
## Call:
## adonis(formula = distance(subs4, method = "bray") ~ cycle, data = metadata,      permutations = 999)
##
```

```
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## cycle      2   0.51429 0.257144  5.1291 0.32818  0.003 **
## Residuals 21   1.05282 0.050134         0.67182
## Total     23   1.56711                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All treatments differ significantly across cycles

test spread of variance of groups

use betadisper test to test for that

```
set.seed(53)
test<-ps1
test2 <- transform_sample_counts(test, function(OTU) {OTU / sum(OTU)})

cycle1 = subset_samples(test2, cycle == "cycle_one")
cycle1 <- prune_taxa(taxa_sums(cycle1) > 0, cycle1)

cycle2 = subset_samples(test2, cycle == "cycle_two")
cycle2 <- prune_taxa(taxa_sums(cycle2) > 0, cycle2)

cycle3_b = subset_samples(test2, cycle == "cycle_three_before_stress")
cycle3_b <- prune_taxa(taxa_sums(cycle3_b) > 0, cycle3_b)

#testing treatment
d_cycle1 = distance(cycle1, "bray")
df_cycle1 = as(sample_data(cycle1), "data.frame")
df_cycle1$treatment3 <- factor(df_cycle1$treatment3 , levels=c("Control", "Chlorothalonil", "Glyphosate
groups <- df_cycle1[["treatment3"]]
beta <- betadisper(d_cycle1, df_cycle1$treatment3)
permutest(beta,pairwise = TRUE, permutations = 999)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq  Mean Sq     F N.Perm Pr(>F)
## Groups     3 0.12856 0.042852 4.266    999  0.016 *
## Residuals 35 0.35157 0.010045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                    Control Chlorothalonil Glyphosate Tetracycline
## Control                       0.4190000  0.1080000        0.007
## Chlorothalonil 0.4157655                 0.3450000        0.008
## Glyphosate     0.1177930      0.3295631                   0.016
## Tetracycline   0.0044018      0.0040197  0.0173897
```

```
anova(betadisper(d_cycle1, groups))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value  Pr(>F)
## Groups     3 0.12856 0.042852   4.266 0.01142 *
## Residuals 35 0.35157 0.010045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod.HSD <- TukeyHSD(beta,conf.level = 0.95)
mod.HSD
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                    diff         lwr          upr     p adj
## Chlorothalonil-Control       0.02676925 -0.08357872  0.137117211 0.9133487
## Glyphosate-Control           0.07354989 -0.03679808  0.183897850 0.2915645
## Tetracycline-Control        -0.15206757 -0.32654302  0.022407884 0.1060221
## Glyphosate-Chlorothalonil    0.04678064 -0.06356733  0.157128605 0.6656851
## Tetracycline-Chlorothalonil -0.17883681 -0.35331227 -0.004361362 0.0428096
## Tetracycline-Glyphosate     -0.22561745 -0.40009291 -0.051142001 0.0069775
```

```
d_cycle2 = distance(cycle2, "bray")
df_cycle2 = as(sample_data(cycle2), "data.frame")
df_cycle2$treatment3 <- factor(df_cycle2$treatment3 , levels=c("Control", "Chlorothalonil", "Glyphosate
groups <- df_cycle2[["treatment3"]]
beta <- betadisper(d_cycle2, df_cycle2$treatment3)
permutest(beta,pairwise = TRUE, permutations = 99)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 99
##
## Response: Distances
##           Df  Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     3 0.02853 0.0095114 0.6322     99   0.59
## Residuals 43 0.64693 0.0150448
##
## Pairwise comparisons:
```

```
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                Control Chlorothalonil Glyphosate Tetracycline
## Control                       0.91000    0.44000         0.53
## Chlorothalonil 0.94337                   0.35000         0.65
## Glyphosate     0.46170        0.27741                    0.12
## Tetracycline   0.66557        0.62562    0.11957
```

```
anova(betadisper(d_cycle2, groups))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq   Mean Sq F value Pr(>F)
## Groups     3 0.02853 0.0095114  0.6322 0.5983
## Residuals 43 0.64693 0.0150448
```

```
mod.HSD <- TukeyHSD(beta)
mod.HSD
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                      diff         lwr        upr     p adj
## Chlorothalonil-Control      -0.004100517 -0.13792105 0.12972001 0.9997999
## Glyphosate-Control           0.044085522 -0.09274259 0.18091363 0.8247329
## Tetracycline-Control        -0.024768446 -0.15858897 0.10905208 0.9598636
## Glyphosate-Chlorothalonil    0.048186039 -0.08864207 0.18501415 0.7829567
## Tetracycline-Chlorothalonil -0.020667929 -0.15448846 0.11315260 0.9759900
## Tetracycline-Glyphosate     -0.068853968 -0.20568207 0.06797414 0.5401612
```

```
d_cycle3_b = distance(cycle3_b, "bray")
df_cycle3_b = as(sample_data(cycle3_b), "data.frame")
df_cycle3_b$treatment3 <- factor(df_cycle3_b$treatment3 , levels=c("Control", "Chlorothalonil", "Glyphos
groups <- df_cycle3_b[["treatment3"]]
beta <- betadisper(d_cycle3_b, df_cycle3_b$treatment3)
permutest(beta,pairwise = TRUE, permutations = 99)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 99
##
## Response: Distances
##           Df  Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups     3 0.02830 0.009432 0.8083     99    0.6
## Residuals 50 0.58346 0.011669
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                  Control Chlorothalonil Glyphosate Tetracycline
## Control                        0.510000   0.670000         0.41
## Chlorothalonil 0.416346                   0.650000         0.02
## Glyphosate     0.732130       0.641863                     0.28
## Tetracycline   0.375692       0.019867   0.246354
```

```
anova(betadisper(d_cycle3_b, groups))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups     3 0.02830 0.009432  0.8083 0.4953
## Residuals 50 0.58346 0.011669
```

```
mod.HSD <- TukeyHSD(beta)
mod.HSD
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                    diff         lwr        upr     p adj
## Chlorothalonil-Control      -0.03526071 -0.14575879 0.07523736 0.8311939
## Glyphosate-Control          -0.01645136 -0.12694944 0.09404672 0.9787710
## Tetracycline-Control         0.04062639 -0.06987169 0.15112446 0.7630330
## Glyphosate-Chlorothalonil    0.01880935 -0.11652260 0.15414131 0.9825928
## Tetracycline-Chlorothalonil  0.07588710 -0.05944485 0.21121905 0.4509069
## Tetracycline-Glyphosate      0.05707775 -0.07825420 0.19240970 0.6785135
```