ITS script codes- Dada2 and phyloseq and stats analysis

2023-03-08

Loading libraries

```
#library("dada2")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package ’ggplot2’ was built under R version 4.1.3
```

```
## Warning: package ’tibble’ was built under R version 4.1.3
```

```
## Warning: package ’dplyr’ was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(phyloseq)
library(rstatix)
```

```
## Warning: package ’rstatix’ was built under R version 4.1.3
```

```
##
## Attaching package: ’rstatix’
```

```
## The following object is masked from ’package:stats’:
##
##     filter
```

Load in global environment. This was done so I don’t have to re-run all the preprocessing code which can take a long time

```
load(file = "ITS.Rdata")
```

Defining path with fastq files

```
#path <- "C:/Users/Asha Mohamed/Desktop/sequencingdata"
#list.files(path)
```

Read in the names of the fastq files, and perform some string manipulation to get matched lists of the forward and reverse fastq files

```
#fnFs <- sort(list.files(path, pattern="_ITS_R1_trimmed.fq", full.names = TRUE))
#fnRs <- sort(list.files(path, pattern="_ITS_R2_trimmed.fq", full.names = TRUE))
```

Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq

```
#sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
```

Inspect read quality profiles

```
#plotQualityProfile(fnFs[1:2])
#plotQualityProfile(fnRs[1:2])
```

Filter and trim

```
## Place filtered files in filtered/ subdirectory
#filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
#filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
#names(filtFs) <- sample.names
#names(filtRs) <- sample.names
## Parameters to be filtered
#out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs,maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,compress=
#head(out)
```

Learn the Error Rates

```
#errF <- learnErrors(filtFs, multithread=TRUE)
#errR <- learnErrors(filtRs, multithread=TRUE)
#plotErrors(errF, nominalQ=TRUE)
```

Sample Inference

```
#dadaFs <- dada(filtFs, err=errF, multithread=TRUE)
#dadaRs <- dada(filtRs, err=errR, multithread=TRUE)
```

Inspecting the returned dada-class object:

```
#dadaFs[[1]]
```

Merge paired reads

```
#mergers <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE)
```

Inspect the merger data.frame from the first sample

```
#head(mergers[[1]])
```

Construct sequence table

```
#seqtab <- makeSequenceTable(mergers)
#dim(seqtab)
```

Inspect distribution of sequence lengths

```
#table(nchar(getsequences(seqtab)))
```

Remove chimeras

```
#seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)
#dim(seqtab.nochim)
#sum(seqtab.nochim)/sum(seqtab)
```

Track reads through the pipeline

```
#getN <- function(x) sum(getUniques(x))
#track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN), rowSums(seqtab.
#colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
#rownames(track) <- sample.names
#head(track)
```

Assign taxonomy

```
#taxa <- assignTaxonomy(seqtab.nochim, "C:/Users/Asha Mohamed/Desktop/sequencingdata/sh_general_release
```

Inspect the taxonomic assignments

```
#taxa.print <- taxa # Removing sequence rownames for display only
#rownames(taxa.print) <- NULL
#head(taxa.print)
```

Construct a simple sample data.frame from the information encoded in the filenames

```
#samples.out <- rownames(seqtab.nochim)
#setwd("C:/Users/Asha Mohamed/Desktop/sequencingdata")
#read.csv("sample_metadata.csv")
#samplesITS <-read.csv("sample_metadata.csv")
```

Import meta data

```
#meta<- read.csv("sample_metadata.csv", header = T, row.names=1)
#meta<- sample_data(meta)
```

Construct a phyloseq object directly from the dada2 outputs

```
#ps <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE),sample_names(samplesITS),tax_table(taxa),

#dna <- Biostrings::DNAStringSet(taxa_names(ps))
#names(dna) <- taxa_names(ps)
#ps <- merge_phyloseq(ps, dna)
#taxa_names(ps) <- paste0("ASV", seq(ntaxa(ps)))
#ps
```

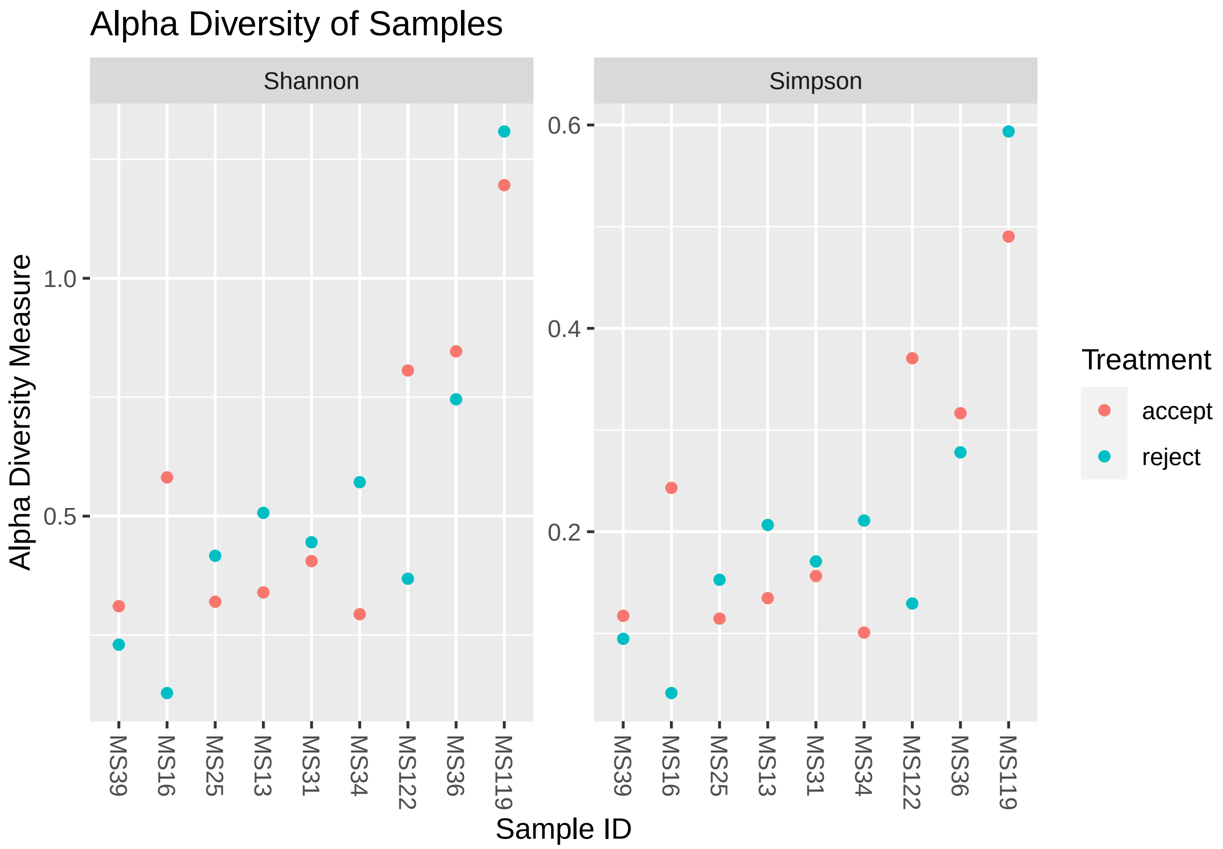
Remove chloroplasts and mitochondria

```
ps<- ps %>% subset_taxa(Family!= "Mitochondria" | is.na(Family) & Order!= "Chloroplast" | is.na(Order))
```

Alpha diversity on raw count data

```
#Visualize alpha-diversity
plot_richness(ps, measures=c("Shannon", "Simpson"), x="Pair", color= "Treatment", title = "Alpha Divers
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```



Transform sample counts to relative abundance

```r
ps_relative<- transform_sample_counts(ps, function(x) x/sum(x))
```

Normalize sample counts

```r
ps_normalized<- rarefy_even_depth(ps, rngseed = 19)
```

```
## ‘set.seed(19)‘ was used to initialize repeatable random subsampling.
```

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

```
## Try ‘set.seed(19); .Random.seed‘ for the full vector
```

```
## ...
```

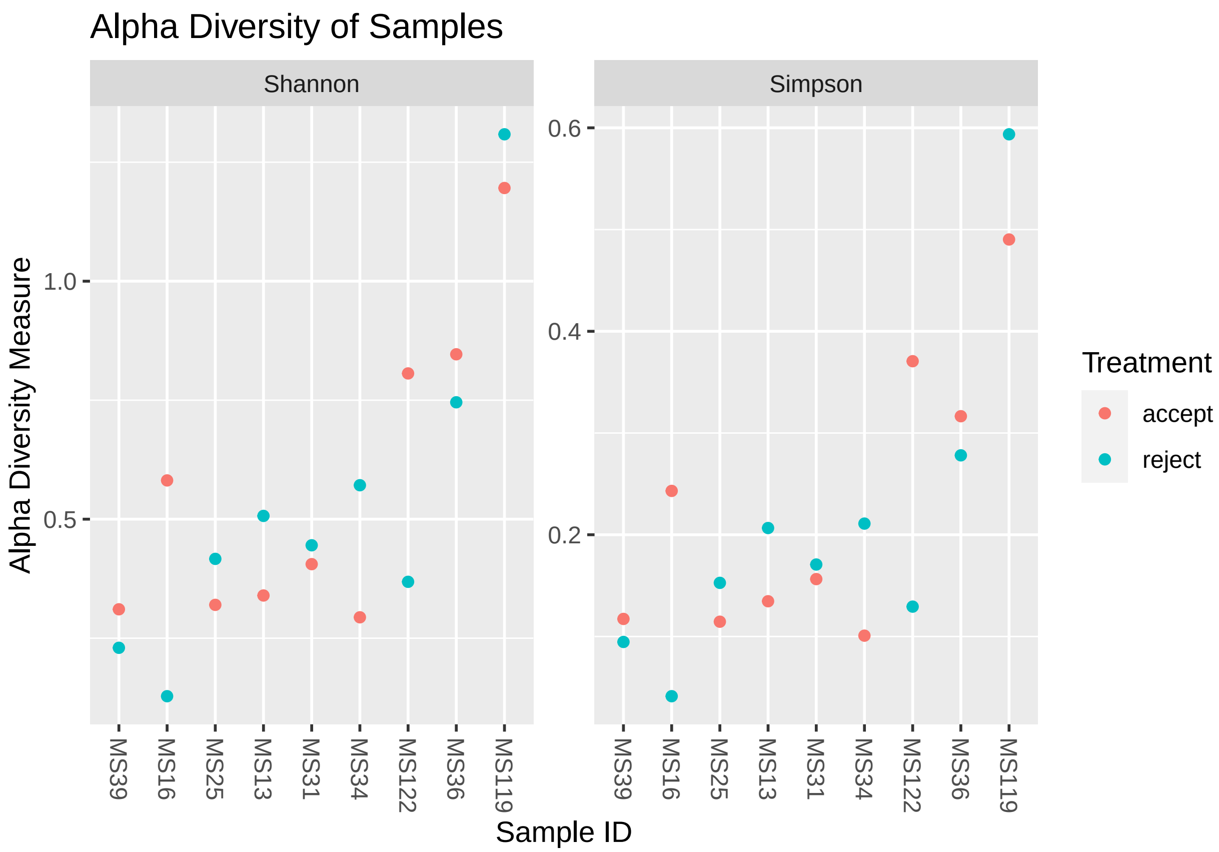
```
## 1OTUs were removed because they are no longer
## present in any sample after random subsampling
```

```
## ...
```

Alpha diversity on relative count data

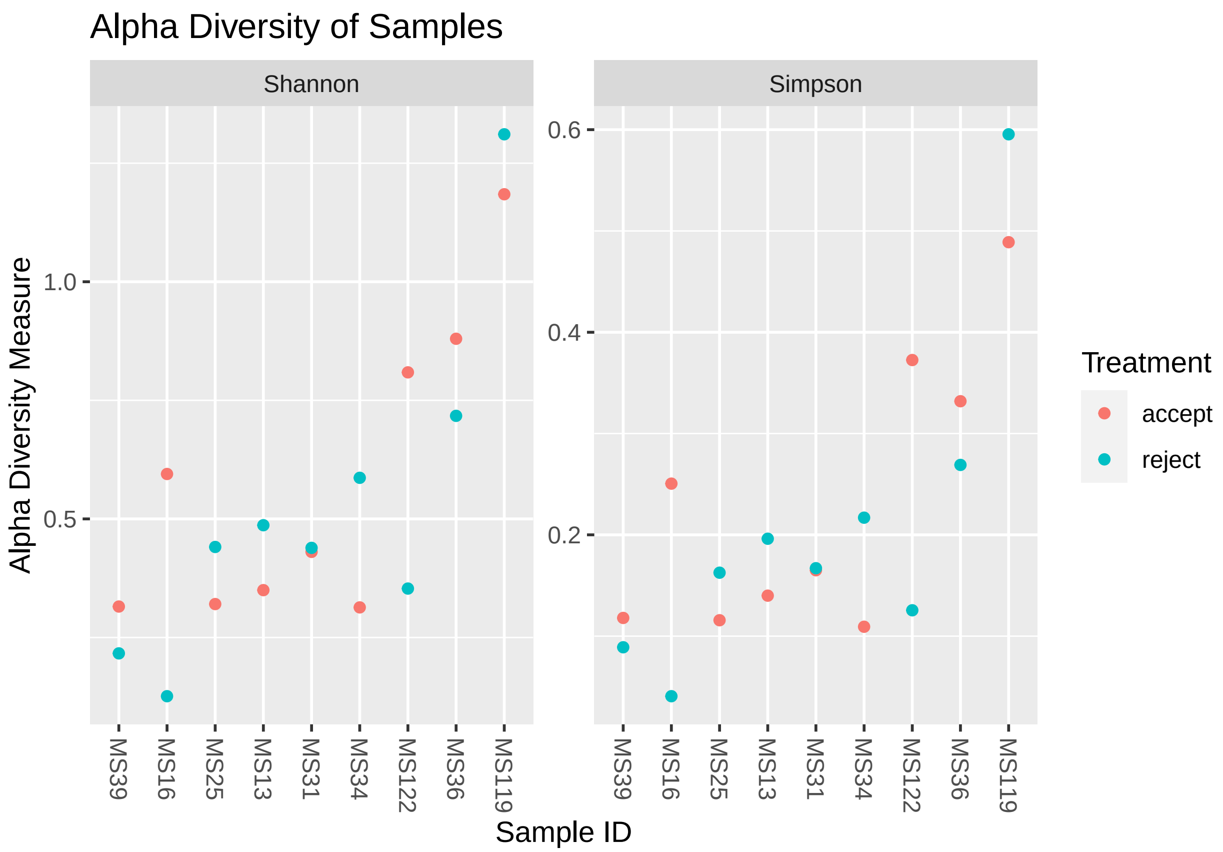
```r
#Visualize alpha-diversity (relative counts)
plot_richness(ps_relative, measures=c("Shannon", "Simpson"), x="Pair", color= "Treatment", title = "Alpl
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```



Alpha diversity on normalized count data

```
#Visualize alpha-diversity (normalized counts)
plot_richness(ps_normalized, measures=c("Shannon", "Simpson"), x="Pair", color= "Treatment", title = "Al
```



```
#stats on alpha diversity (normalized)
alpha_norm<- estimate_richness(ps_normalized)

alpha_norm$Treatment<- meta$Treatment

alpha_model_norm<- lm(Shannon ~ Treatment, data= alpha_norm)
alpha_model2_norm<- anova_test(alpha_model_norm)
```

```
## Coefficient covariances computed by hccm()
```

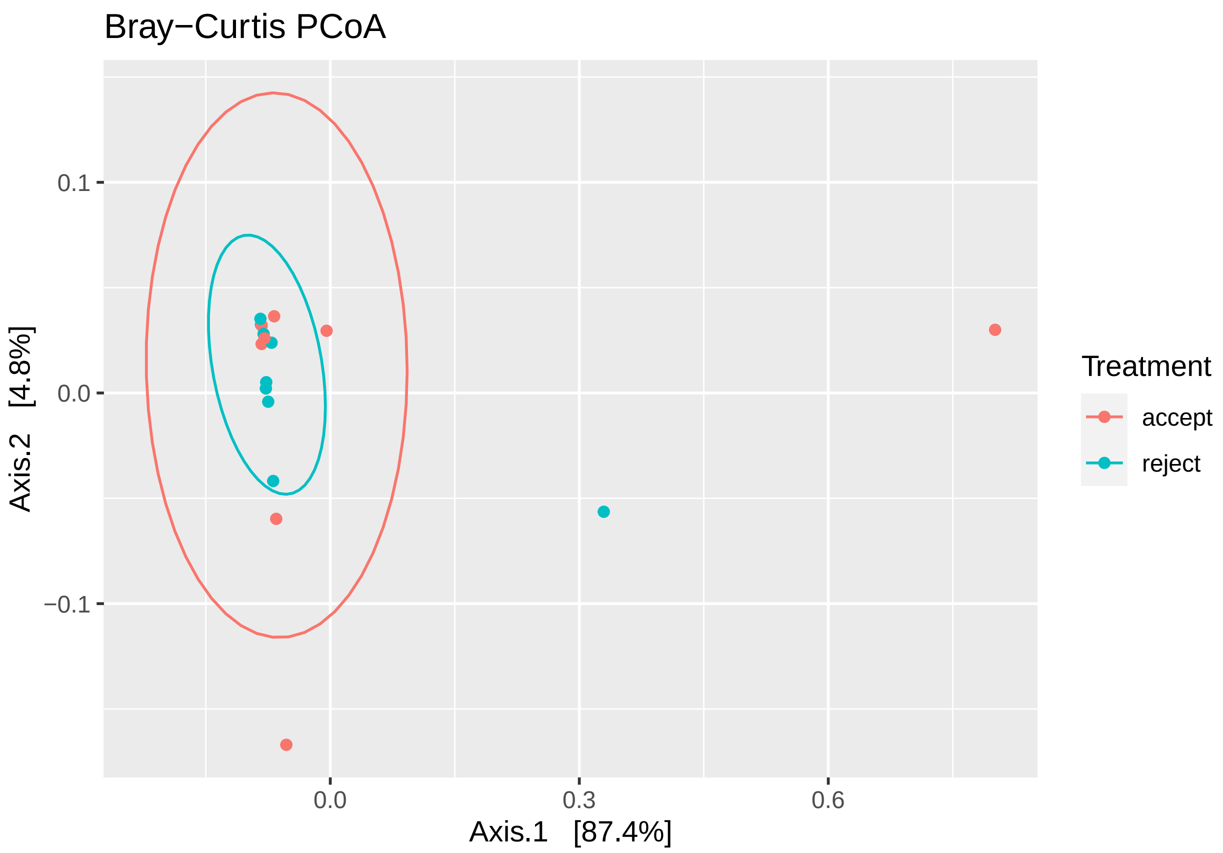
```
alpha_model2_norm
```

```
## ANOVA Table (type II tests)
##
##      Effect DFn DFd     F     p p<.05   ges
## 1 Treatment   1  16 0.138 0.716       0.009
```

Beta diversity on relative count data

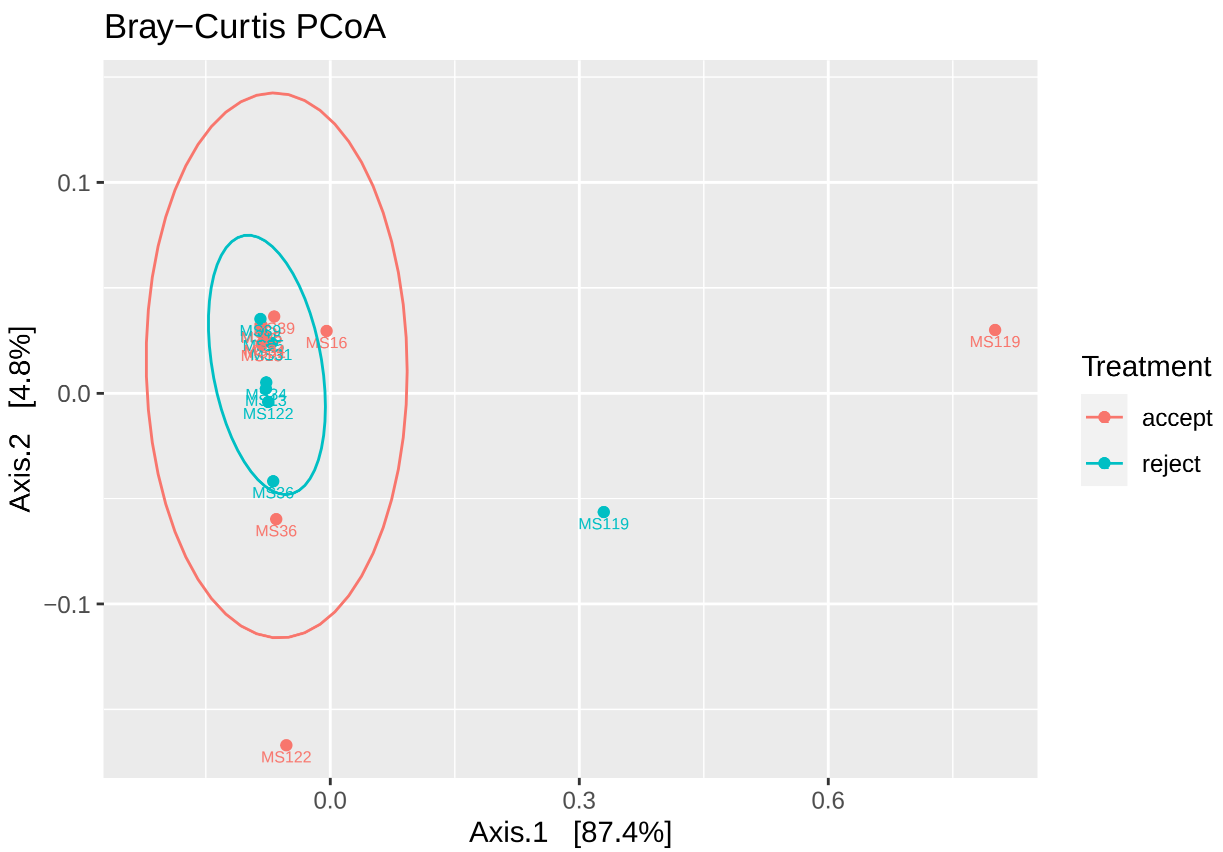
```
PCoA_relative<- ordinate(ps_relative, method="PCoA", distance="bray")

plot_ordination(ps_relative, PCoA_relative, color = "Treatment", title="Bray-Curtis PCoA") + stat_ellips
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```



```
#plot with sample labels
plot_ordination(ps_relative, PCoA_relative, color = "Treatment", title="Bray-Curtis PCoA", label = "Pai
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```



```
#PERMANOVA on relative counts beta diversity
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 4.1.3

## Loading required package: permute

## Warning: package 'permute' was built under R version 4.1.3

## Loading required package: lattice

## This is vegan 2.6-2
```

```
distance_matrix_relative<- phyloseq::distance(ps_relative, method = "bray")

permanova_relative<- adonis2(distance_matrix_relative ~ phyloseq::sample_data(ps_relative)$Treatment, m
permanova_relative
```

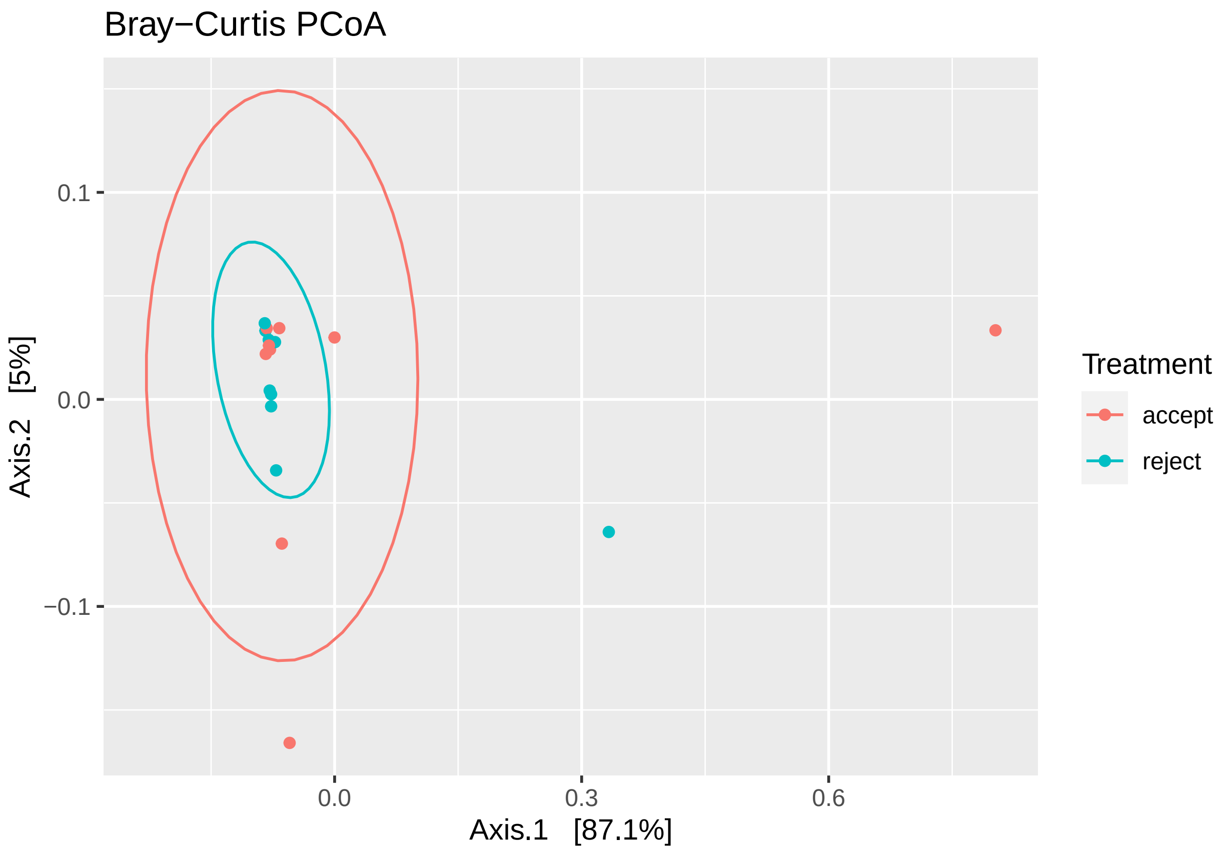
```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
```

```
## adonis2(formula = distance_matrix_relative ~ phyloseq::sample_data(ps_relative)$Treatment, method =
##                                              Df SumOfSqs      R2     F Pr(>F)
## phyloseq::sample_data(ps_relative)$Treatment  1  0.02651 0.02771 0.456  0.569
## Residual                                     16  0.93023 0.97229
## Total                                        17  0.95674 1.00000
```

Beta diversity on normalized count data

```
PCoA_norm<- ordinate(ps_normalized, method="PCoA", distance="bray")

plot_ordination(ps_normalized, PCoA_norm, color = "Treatment", title="Bray-Curtis PCoA") + stat_ellipse
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```



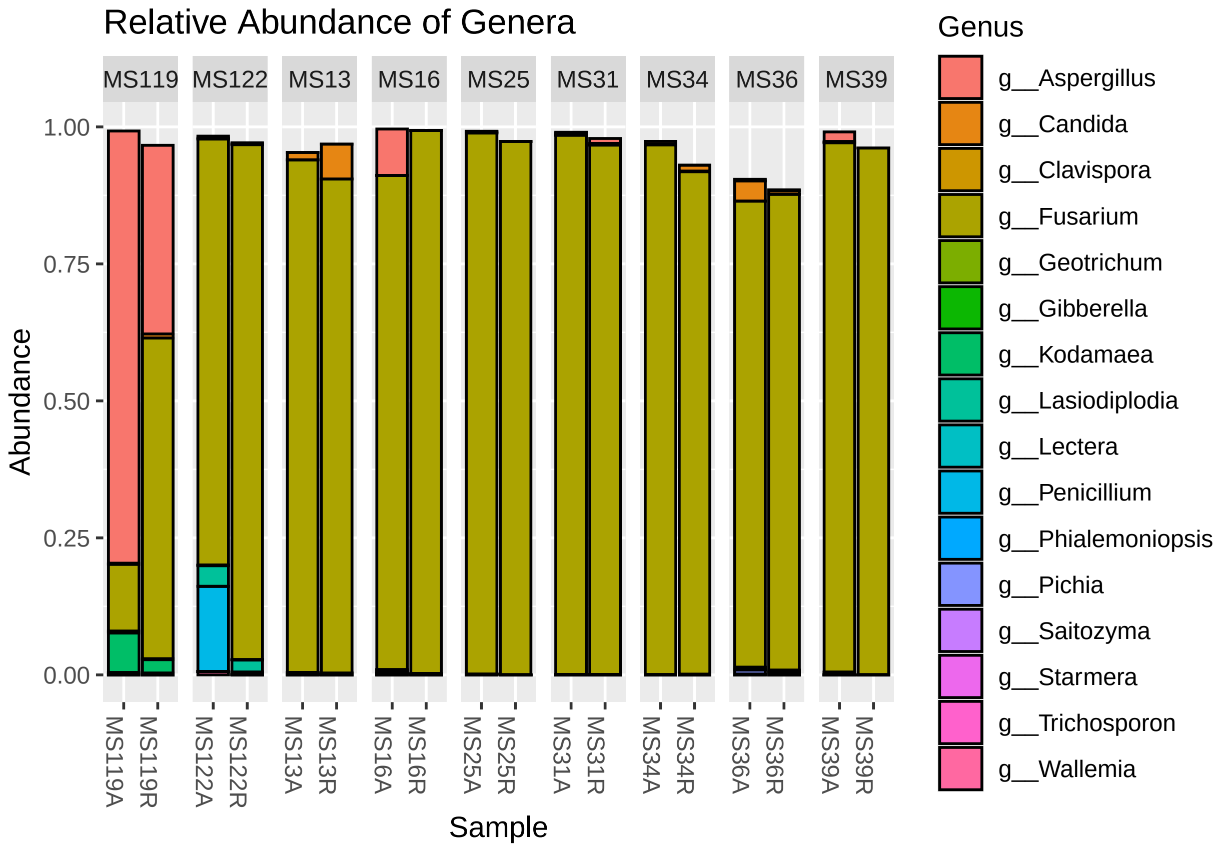
```
#PERMANOVA on normalized counts beta diversity
distance_matrix_norm<- phyloseq::distance(ps_normalized, method = "bray")

permanova_norm<- adonis2(distance_matrix_norm ~ phyloseq::sample_data(ps_normalized)$Treatment, method =
permanova_norm
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
```

```
##
## adonis2(formula = distance_matrix_norm ~ phyloseq::sample_data(ps_normalized)$Treatment, method = "b
##                                                 Df SumOfSqs      R2      F
## phyloseq::sample_data(ps_normalized)$Treatment  1  0.02685 0.02778 0.4572
## Residual                                       16  0.93966 0.97222
## Total                                          17  0.96651 1.00000
##                                                Pr(>F)
## phyloseq::sample_data(ps_normalized)$Treatment  0.588
## Residual
## Total
```

Visualize genus (relative abundance)

```
tax_glom_genus<- tax_glom(ps_relative, taxrank = "Genus")
plot_bar(tax_glom_genus, fill = "Genus", title = "Relative Abundance of Genera") + facet_grid(~Pair, sc
```



Write genus abundance data into a csv file

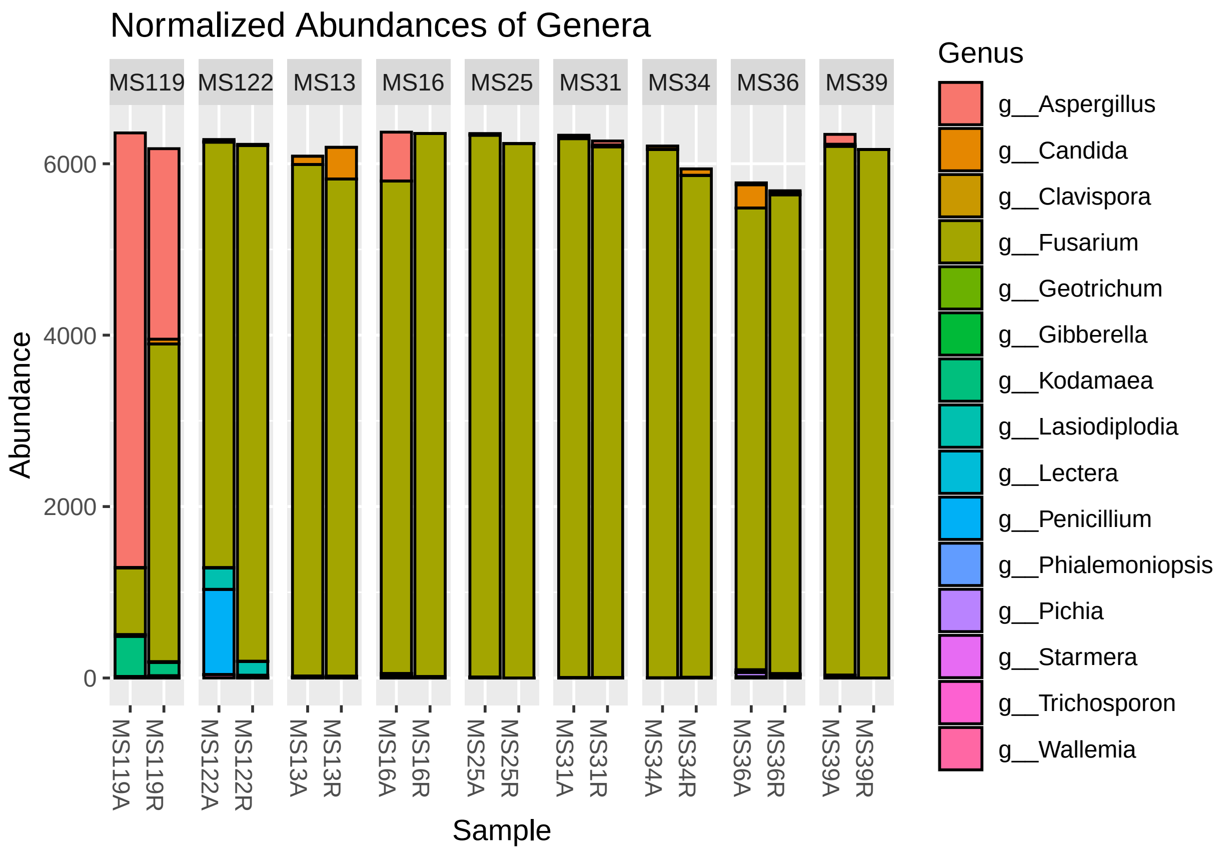
```
ps_genus_results<- psmelt(tax_glom_genus)
ps_genus_results<- ps_genus_results %>%
  group_by(Sample,Genus) %>%
  summarise(abundance =sum(Abundance))
```

```
## `summarise()` has grouped output by 'Sample'. You can override using the
## `.groups` argument.
```

```r
write.csv(ps_genus_results, "ps_genus_results.csv")
```

Visualize genus (normalized data)

```r
tax_glom_genus_normal<- tax_glom(ps_normalized, taxrank = "Genus")
plot_bar(tax_glom_genus_normal, fill = "Genus", title = "Normalized Abundances of Genera") + facet_grid
```



Write csv of phyloseq object

```r
psmelt_ps<- psmelt(ps)
write.csv(psmelt_ps, "psmelt_ps.csv")
```

ALDEx2 and LEFSE stats analysis

```r
library(microbiomeMarker)
```

```
## Registered S3 method overwritten by 'ggtree':
##   method      from
##   identify.gg ggfun
```

```
##
## Attaching package: 'microbiomeMarker'
## The following object is masked from ’package:phyloseq’:
##
##     plot_heatmap
```

```
##raw data
#aldex (ANOVA like differential expression)
aldex<- run_aldex(ps, group = "Treatment", taxa_rank = "all", transform = "identity", norm = "none", me
```

```
## operating in serial mode
```

```
## computing center with all features
```

```
## New names:
## * ‘‘ -> ‘...1‘
## * ‘‘ -> ‘...2‘
## * ‘‘ -> ‘...3‘
## * ‘‘ -> ‘...4‘
## * ‘‘ -> ‘...5‘
## * ‘‘ -> ‘...6‘
## * ‘‘ -> ‘...7‘
## * ‘‘ -> ‘...8‘
## * ‘‘ -> ‘...9‘
## * ‘‘ -> ‘...10‘
## * ‘‘ -> ‘...11‘
## * ‘‘ -> ‘...12‘
## * ‘‘ -> ‘...13‘
## * ‘‘ -> ‘...14‘
## * ‘‘ -> ‘...15‘
## * ‘‘ -> ‘...16‘
## * ‘‘ -> ‘...17‘
## * ‘‘ -> ‘...18‘
## * ‘‘ -> ‘...19‘
## * ‘‘ -> ‘...20‘
## * ‘‘ -> ‘...21‘
## * ‘‘ -> ‘...22‘
## * ‘‘ -> ‘...23‘
## * ‘‘ -> ‘...24‘
## * ‘‘ -> ‘...25‘
## * ‘‘ -> ‘...26‘
## * ‘‘ -> ‘...27‘
## * ‘‘ -> ‘...28‘
## * ‘‘ -> ‘...29‘
## * ‘‘ -> ‘...30‘
## * ‘‘ -> ‘...31‘
## * ‘‘ -> ‘...32‘
## * ‘‘ -> ‘...33‘
## * ‘‘ -> ‘...34‘
## * ‘‘ -> ‘...35‘
## * ‘‘ -> ‘...36‘
## * ‘‘ -> ‘...37‘
## * ‘‘ -> ‘...38‘
## * ‘‘ -> ‘...39‘
## * ‘‘ -> ‘...40‘
## * `` -> `...41`
## * `` -> `...42`
## * `` -> `...43`
## * `` -> `...44`
## * `` -> `...45`
## * `` -> `...46`
## * `` -> `...47`
## * `` -> `...48`
## * `` -> `...49`
## * `` -> `...50`
## * `` -> `...51`
## * `` -> `...52`
## * `` -> `...53`
## * `` -> `...54`
## * `` -> `...55`
## * `` -> `...56`
## * `` -> `...57`
## * `` -> `...58`
## * `` -> `...59`
## * `` -> `...60`
## * `` -> `...61`
## * `` -> `...62`
## * `` -> `...63`
## * `` -> `...64`
## * `` -> `...65`
## * `` -> `...66`
## * `` -> `...67`
## * `` -> `...68`
## * `` -> `...69`
## * `` -> `...70`
## * `` -> `...71`
## * `` -> `...72`
## * `` -> `...73`
## * `` -> `...74`
## * `` -> `...75`
## * `` -> `...76`
## * `` -> `...77`
## * `` -> `...78`
## * `` -> `...79`
## * `` -> `...80`
## * `` -> `...81`
## * `` -> `...82`
## * `` -> `...83`
## * `` -> `...84`
## * `` -> `...85`
## * `` -> `...86`
## * `` -> `...87`
## * `` -> `...88`
## * `` -> `...89`
## * `` -> `...90`
## * `` -> `...91`
## * `` -> `...92`
## * `` -> `...93`
## * `` -> `...94`
## * `` -> `...95`
## * `` -> `...96`
## * `` -> `...97`
## * `` -> `...98`
## * `` -> `...99`
## * `` -> `...100`
## * `` -> `...101`
## * `` -> `...102`
## * `` -> `...103`
## * `` -> `...104`
## * `` -> `...105`
## * `` -> `...106`
## * `` -> `...107`
## * `` -> `...108`
## * `` -> `...109`
## * `` -> `...110`
## * `` -> `...111`
## * `` -> `...112`
## * `` -> `...113`
## * `` -> `...114`
## * `` -> `...115`
## * `` -> `...116`
## * `` -> `...117`
## * `` -> `...118`
## * `` -> `...119`
## * `` -> `...120`
## * `` -> `...121`
## * `` -> `...122`
## * `` -> `...123`
## * `` -> `...124`
## * `` -> `...125`
## * `` -> `...126`
## * `` -> `...127`
## * `` -> `...128`
```

```
## Warning: No marker was identified
```

```
#lefse (linear discriminant analysis effect size)
lefse<- run_lefse(ps, group = "Treatment", taxa_rank = "all", transform = "identity", norm = "none")

##normalized data
#aldex
aldex_norm<- run_aldex(ps_normalized, group = "Treatment", taxa_rank = "all", transform = "identity", n
```

```
## operating in serial mode
## computing center with all features
## New names:
```

```
## Warning: No marker was identified
```

```
#lefse
lefse_norm<- run_lefse(ps_normalized, group = "Treatment", taxa_rank = "all", transform = "identity", n
## Warning: No marker was identified
```

```
##relative abundances
#aldex
aldex_rel<- run_aldex(ps_relative, group = "Treatment", taxa_rank = "all", transform = "identity", norm
```

```
## operating in serial mode
```

```
## Warning: Not all reads are integers, the reads are ceiled to integers.
##    Raw reads is recommended from the ALDEx2 paper.
```

```
## operating in serial mode
## computing center with all features
## New names:
```

```
## Warning: No marker was identified
```

```
#lefse
lefse_rel<- run_lefse(ps_relative, group = "Treatment", taxa_rank = "all", transform = "identity", norm
```

```
## Warning: No marker was identified
```

ANOVA of Aspergillus and Fusarium between accept and reject samples

```
##relative data
ps_genus_results<- psmelt(tax_glom_genus)
ps_genus_results<- ps_genus_results %>%
  group_by(Sample,Genus) %>%
  summarise(abundance =sum(Abundance))
```

```
## `summarise()` has grouped output by 'Sample'. You can override using the
## `.groups` argument.
```

```
#aspergillus
ps_relative_aspergillus<- filter(ps_genus_results, Genus == "g__Aspergillus")
ps_relative_aspergillus$Treatment<- meta$Treatment

asper_model_rel<- lm(abundance ~ Treatment, data= ps_relative_aspergillus)
asper_model2_rel<- anova_test(asper_model_rel)
```

```
## Coefficient covariances computed by hccm()
```

```
asper_model2_rel
```

```
## ANOVA Table (type II tests)
##
##      Effect DFn DFd     F     p p<.05   ges
## 1 Treatment   1  16 0.406 0.533       0.025
```

```
#fusarium
ps_relative_fusarium<- filter(ps_genus_results, Genus == "g__Fusarium")
ps_relative_fusarium$Treatment<- meta$Treatment

fus_model_rel<- lm(abundance ~ Treatment, data= ps_relative_fusarium)
fus_model2_rel<- anova_test(fus_model_rel)
```

```
## Coefficient covariances computed by hccm()
```

```
fus_model2_rel
```

```
## ANOVA Table (type II tests)
##
##      Effect DFn DFd     F    p p<.05   ges
## 1 Treatment   1  16 0.454 0.51       0.028
```

```
##normalized data
ps_genus_results_norm<- psmelt(tax_glom_genus_normal)
ps_genus_results_norm<- ps_genus_results_norm %>%
  group_by(Sample,Genus) %>%
  summarise(abundance =sum(Abundance))
```

```
## ‘summarise()‘ has grouped output by ’Sample’. You can override using the
## ‘.groups‘ argument.
```

```
#aspergillus
ps_norm_aspergillus<- filter(ps_genus_results_norm, Genus == "g__Aspergillus")
ps_norm_aspergillus$Treatment<- meta$Treatment

asper_model_norm<- lm(abundance ~ Treatment, data= ps_norm_aspergillus)
asper_model2_norm<- anova_test(asper_model_norm)
```

```
## Coefficient covariances computed by hccm()
```

```
asper_model2_norm
```

```
## ANOVA Table (type II tests)
##
##      Effect DFn DFd     F    p p<.05   ges
## 1 Treatment   1  16 0.413 0.53       0.025
```

```
#fusarium
ps_norm_fusarium<- filter(ps_genus_results_norm, Genus == "g__Fusarium")
ps_norm_fusarium$Treatment<- meta$Treatment

fus_model_norm<- lm(abundance ~ Treatment, data= ps_norm_fusarium)
fus_model2_norm<- anova_test(fus_model_norm)
```

```
## Coefficient covariances computed by hccm()
```

```
fus_model2_norm
```

```
## ANOVA Table (type II tests)
##
##      Effect DFn DFd     F     p p<.05  ges
## 1 Treatment   1  16 0.501 0.489       0.03
```