

First experimental evidence for active farming in ambrosia beetles and strong heredity of garden microbiomes - Bacterial Analysis

Janina M.C. Diehl

08.07.2022

Data Preparation

load required packages

```
library(lme4)
library(permute)
library(lattice)
library(vegan)
library(phyloseq)
library(ggplot2)
library(dplyr)
library(scales)
library(grid)
library(DHARMa)
library(ggeffects)
library(glmmTMB)
library(lmerTest)
library(emmeans)
library(sjPlot)
library(fitdistrplus)
library(GLMMadaptive)
library(microbiome)
library(microbiomeutilities)
library(knitr)
library(ggpubr)
library(doBy)
library(performance)
library(see)
library(patchwork)
library(pairwiseAdonis)
library(cowplot)
library(multcomp)
library(car)
library(forcats)
library(ggrepel)
library(tidyverse)
```

loading the data files for bacteria

```
data16S <- otu_table(read.table("16S_Rem_zotu_table.txt", sep="\t", header=T, row.names=1, check.names =  
tax16S <- tax_table(as.matrix(read.table("16S_Rem_zotus.tax.txt", sep="\t", header=T, row.names=1, fill=NA)  
datasample16S <- sample_data(read.table("16S_map_removal.txt", sep="\t", header=T, row.names=1)))
```

merge data into phyloseq object

```
(all16S <- merge_phyloseq(data16S, tax16S, datasample16S))
```

```
## phyloseq-class experiment-level object  
## otu_table() OTU Table: [ 365 taxa and 57 samples ]  
## sample_data() Sample Data: [ 57 samples by 16 sample variables ]  
## tax_table() Taxonomy Table: [ 365 taxa by 8 taxonomic ranks ]
```

copy taxonomic classification in tax_table columns with gaps and add "__spc"

```
dataset.16S = subset_taxa(all16S, (Domain == "d:Bacteria"))  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Phylum"] == "", "Phylum"] <- paste(tax_table(dataset.16S)[tax_  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Class"] == "", "Class"] <- paste(tax_table(dataset.16S)[tax_  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Order"] == "", "Order"] <- paste(tax_table(dataset.16S)[tax_  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Family"] == "", "Family"] <- paste(tax_table(dataset.16S)[tax_  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Genus"] == "", "Genus"] <- paste(tax_table(dataset.16S)[tax_  
tax_table(dataset.16S)[tax_table(dataset.16S)[, "Species"] == "", "Species"] <- paste(tax_table(dataset.16S)[tax_
```

start filtering out all ZOTUs that were only assigned to Domain (Bacteria and Archaea), as well as Chloroplasts

```
dataset.16S.ordi = subset_taxa(dataset.16S, (Domain == "d:Bacteria" | Domain == "d:Archaea"))  
dataset.16S.ordi = subset_taxa(dataset.16S.ordi, Phylum != "d:Bacteria_spc")  
dataset.16S.ordi = subset_taxa(dataset.16S.ordi, Phylum != "d:Archaea_spc")  
dataset.16S.ordi = subset_taxa(dataset.16S.ordi, Class != "c:Chloroplast")
```

controlled for most abundant ZOTUS in the dataset with Bacteria sp. and Archaea sp. still included in
https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastn&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome ==> mitochondrial DNA

Zotu1 -> Leptographium (Ophiostomataceae) max/total score: 340, Query Cover: 100%, e-value: 2e-89, Per. Ident 93.81%

Zotu3 -> Leptographium (Ophiostomataceae) max/total score: 274, Query Cover: 100%, e-value: 2e-69, Per. Ident 88.60%

Zotu4 -> Ophiostoma (Query Cover : 100%, Per. Ident: 89.96%) or Grosmannia (Query Cover: 93%, Per. Ident: 91.51%)

Zotu6 -> Hydropisphaera (Trichoderma, Fusarium close) max/total score: 316, Query Cover: 100%, e-value: 3e-82, Per. Ident 92.44%

Zotu7 -> Wolbachia rRNA max/total score: 468, Query Cover: 100%, e-value: 8e-128, Per. Ident 100%

Zotu11 -> Clematis sp. or Ranunculus sp. max/total score: 464, Query Cover: 100%, e-value: 1e-126, Per. Ident 100%

Zotu12 -> Cannabis max/total score: 464, Query Cover: 100%, e-value: 1e-126, Per. Ident 100%

note: we need to exclude these Zotus since they are not bacterial species (except Wolbachia), keep this in mind for the fungal analysis

check all the columns for patterns ranging from [a-z] joined by ____ like this [a-z]____ and substitute it with "" i.e. nothing.

```
tax_table(dataset.16S.ordi[, colnames(tax_table(dataset.16S.ordi))] <- gsub(tax_table(dataset.16S.ordi.
tax_table(dataset.16S.ordi[, colnames(tax_table(dataset.16S.ordi))] <- gsub(tax_table(dataset.16S.ordi.
tax_table(dataset.16S.ordi[, colnames(tax_table(dataset.16S.ordi))] <- gsub(tax_table(dataset.16S.ordi.
tax_table(dataset.16S.ordi[, colnames(tax_table(dataset.16S.ordi))] <- gsub(tax_table(dataset.16S.ordi.
tax_table(dataset.16S.ordi) <- tax_table(dataset.16S.ordi)[,2:8]
```

Format the phyloseq object to add the best taxonomy in phyloseq object (tax_table and otu_table).

```
dataset.16S.ordi <- format_to_besthit(dataset.16S.ordi)
taxa_names(dataset.16S.ordi)[1:5]
```

```
## [1] "Zotu2:Pseudoxanthomonas_spadix" "Zotu48:Microbacterium_spc"
## [3] "Zotu21:Microbacterium_pygmaeum" "Zotu66:Sphingobacterium_spc"
## [5] "Zotu260:Gp3_spc"
```

```
colSums(otu_table(dataset.16S.ordi))
```

check the table without Bacteria and Archaea sp. for total reads per sample

##	B0-19a	B0-19b	B0-22a	B0-22b	B0-24	B0-27	B10-01	B10-06
##	17492	17718	1365	2348	1424	803	659	226
##	B10-09a	B10-09b	B10-18	B10-24	B10-25a	B10-25b	B10-26a	B15-01a
##	133	132	175	221	98	202	261	359
##	B15-01b	B15-06	B15-07a	B15-07b	B15-11	B15-20	B15-24	B15-25a
##	306	3589	3152	1184	3184	4624	1181	281
##	B15-25b	B16-01a	B16-04	B16-31a	B16-31b	B20-04a	B20-08a	B20-08b
##	1156	24582	24435	36064	20380	28703	7581	14344
##	B20-11	B23-21	B23-23a	B24-02a	B24-26a	B24-26b	B24-28a	B24-29
##	22250	312	472	21494	44940	9135	14354	174
##	B36-14a	B36-14b	B36-30	B36-31a	B36-32	B39-15	B39-16	B39-17a
##	12948	10866	11903	24530	24335	26401	26823	29121
##	B46-03	B7-07	B7-08a	medium1	medium2	negative1	negative2	standard1
##	15551	19370	12785	3672	6445	2584	3926	10263
##	standard2							
##	9062							

there are now samples that lost a lot of reads due to this filtering step -> not good for later analysis, some samples might get excluded ahead to secure robustness

check controls from dataset: only ten most abundant ZOTUs are picked for visualisation

```
neg.controls<-subset_samples(dataset.16S.ordi, Treatment=="medium" | Treatment=="negative")
sample_names(neg.controls)
```

```
## [1] "medium1" "medium2" "negative1" "negative2"
```

```
neg <- prune_taxa(taxa_sums(neg.controls) > 0, neg.controls)
```

visualisation of negative controls

```
filttaxa <- names (sort(rowSums(otu_table(neg)) ,decreasing=T))[1:10]
tax_table(dataset.16S.ordi)[filttaxa]
```

```
## Taxonomy Table:      [10 taxa by 8 taxonomic ranks]:
##
## Zotu19:Streptomyces_xanthophaeus      "Bacteria" "Actinobacteria"
## Zotu20:Pseudomonas_brenneri           "Bacteria" "Proteobacteria"
## Zotu28:Halomonadaceae_spc             "Bacteria" "Proteobacteria"
## Zotu36:Pelomonas_aquatica             "Bacteria" "Proteobacteria"
## Zotu17:Staphylococcus_aureus          "Bacteria" "Firmicutes"
## Zotu33:Sediminibacterium_spc          "Bacteria" "Bacteroidetes"
## Zotu42:Halomonas_desiderata           "Bacteria" "Proteobacteria"
## Zotu39:Acinetobacter_indicus          "Bacteria" "Proteobacteria"
## Zotu41:Lactobacillus_spc              "Bacteria" "Firmicutes"
## Zotu40:Undibacterium_oligocarboniphilum "Bacteria" "Proteobacteria"
##
## Zotu19:Streptomyces_xanthophaeus      "Actinobacteria"
## Zotu20:Pseudomonas_brenneri           "Gammaproteobacteria"
## Zotu28:Halomonadaceae_spc             "Gammaproteobacteria"
## Zotu36:Pelomonas_aquatica             "Betaproteobacteria"
## Zotu17:Staphylococcus_aureus          "Bacilli"
## Zotu33:Sediminibacterium_spc          "Sphingobacteriia"
## Zotu42:Halomonas_desiderata           "Gammaproteobacteria"
## Zotu39:Acinetobacter_indicus          "Gammaproteobacteria"
## Zotu41:Lactobacillus_spc              "Bacilli"
## Zotu40:Undibacterium_oligocarboniphilum "Betaproteobacteria"
##
## Zotu19:Streptomyces_xanthophaeus      "Actinomycetales"
## Zotu20:Pseudomonas_brenneri           "Pseudomonadales"
## Zotu28:Halomonadaceae_spc             "Oceanospirillales"
## Zotu36:Pelomonas_aquatica             "Burkholderiales"
## Zotu17:Staphylococcus_aureus          "Bacillales"
## Zotu33:Sediminibacterium_spc          "Sphingobacteriales"
## Zotu42:Halomonas_desiderata           "Oceanospirillales"
## Zotu39:Acinetobacter_indicus          "Pseudomonadales"
```

```

## Zotu41:Lactobacillus_spc "Lactobacillales"
## Zotu40:Undibacterium_oligocarboniphilum "Burkholderiales"
## Family
## Zotu19:Streptomyces_xanthophaeus "Streptomycetaceae"
## Zotu20:Pseudomonas_brenneri "Pseudomonadaceae"
## Zotu28:Halomonadaceae_spc "Halomonadaceae"
## Zotu36:Pelomonas_aquatica "Comamonadaceae"
## Zotu17:Staphylococcus_aureus "Staphylococcaceae"
## Zotu33:Sediminibacterium_spc "Chitinophagaceae"
## Zotu42:Halomonas_desiderata "Halomonadaceae"
## Zotu39:Acinetobacter_indicus "Moraxellaceae"
## Zotu41:Lactobacillus_spc "Lactobacillaceae"
## Zotu40:Undibacterium_oligocarboniphilum "Oxalobacteraceae"
## Genus
## Zotu19:Streptomyces_xanthophaeus "Streptomyces"
## Zotu20:Pseudomonas_brenneri "Pseudomonas"
## Zotu28:Halomonadaceae_spc "Halomonadaceae_spc"
## Zotu36:Pelomonas_aquatica "Pelomonas"
## Zotu17:Staphylococcus_aureus "Staphylococcus"
## Zotu33:Sediminibacterium_spc "Sediminibacterium"
## Zotu42:Halomonas_desiderata "Halomonas"
## Zotu39:Acinetobacter_indicus "Acinetobacter"
## Zotu41:Lactobacillus_spc "Lactobacillus"
## Zotu40:Undibacterium_oligocarboniphilum "Undibacterium"
## Species
## Zotu19:Streptomyces_xanthophaeus "Streptomyces_xanthophaeus"
## Zotu20:Pseudomonas_brenneri "Pseudomonas_brenneri"
## Zotu28:Halomonadaceae_spc "Halomonadaceae_spc"
## Zotu36:Pelomonas_aquatica "Pelomonas_aquatica"
## Zotu17:Staphylococcus_aureus "Staphylococcus_aureus"
## Zotu33:Sediminibacterium_spc "Sediminibacterium_spc"
## Zotu42:Halomonas_desiderata "Halomonas_desiderata"
## Zotu39:Acinetobacter_indicus "Acinetobacter_indicus"
## Zotu41:Lactobacillus_spc "Lactobacillus_spc"
## Zotu40:Undibacterium_oligocarboniphilum "Undibacterium_oligocarboniphilum"
## best_hit
## Zotu19:Streptomyces_xanthophaeus "Zotu19:Streptomyces_xanthophaeus"
## Zotu20:Pseudomonas_brenneri "Zotu20:Pseudomonas_brenneri"
## Zotu28:Halomonadaceae_spc "Zotu28:Halomonadaceae_spc"
## Zotu36:Pelomonas_aquatica "Zotu36:Pelomonas_aquatica"
## Zotu17:Staphylococcus_aureus "Zotu17:Staphylococcus_aureus"
## Zotu33:Sediminibacterium_spc "Zotu33:Sediminibacterium_spc"
## Zotu42:Halomonas_desiderata "Zotu42:Halomonas_desiderata"
## Zotu39:Acinetobacter_indicus "Zotu39:Acinetobacter_indicus"
## Zotu41:Lactobacillus_spc "Zotu41:Lactobacillus_spc"
## Zotu40:Undibacterium_oligocarboniphilum "Zotu40:Undibacterium_oligocarboniphilum"

```

```
round(otu_table(dataset.16S.ordi)[filtaxa], digits = 4)
```

```

## OTU Table:          [10 taxa and 57 samples]
##                      taxa are rows
##
##                      B0-19a B0-19b B0-22a B0-22b B0-24 B0-27
## Zotu19:Streptomyces_xanthophaeus      29      0      7      0      2      0
## Zotu20:Pseudomonas_brenneri           8      0     46      0      0      1

```

## Zotu28:Halomonadaceae_spc	10	1	23	0	0	0
## Zotu36:Pelomonas_aquatica	10	0	18	2	4	1
## Zotu17:Staphylococcus_aureus	7	0	7	4	2	0
## Zotu33:Sediminibacterium_spc	12	0	10	0	0	0
## Zotu42:Halomonas_desiderata	6	0	12	0	0	0
## Zotu39:Acinetobacter_indicus	2	0	15	0	2	0
## Zotu41:Lactobacillus_spc	0	0	16	0	1	0
## Zotu40:Undibacterium_oligocarboniphilum	3	0	7	0	0	0
##	B10-01	B10-06	B10-09a	B10-09b	B10-18	
## Zotu19:Streptomyces_xanthophaeus	71	0	0	1	4	
## Zotu20:Pseudomonas_brenneri	68	0	0	4	2	
## Zotu28:Halomonadaceae_spc	25	0	0	0	1	
## Zotu36:Pelomonas_aquatica	12	2	0	0	2	
## Zotu17:Staphylococcus_aureus	14	1	0	0	1	
## Zotu33:Sediminibacterium_spc	16	0	0	0	0	
## Zotu42:Halomonas_desiderata	9	0	0	0	3	
## Zotu39:Acinetobacter_indicus	4	0	0	1	0	
## Zotu41:Lactobacillus_spc	8	0	0	0	0	
## Zotu40:Undibacterium_oligocarboniphilum	14	1	0	0	0	
##	B10-24	B10-25a	B10-25b	B10-26a	B15-01a	
## Zotu19:Streptomyces_xanthophaeus	0	3	1	1	1	
## Zotu20:Pseudomonas_brenneri	0	2	1	0	5	
## Zotu28:Halomonadaceae_spc	0	0	0	0	1	
## Zotu36:Pelomonas_aquatica	3	0	1	2	10	
## Zotu17:Staphylococcus_aureus	3	10	6	3	12	
## Zotu33:Sediminibacterium_spc	10	0	0	1	0	
## Zotu42:Halomonas_desiderata	0	0	2	0	0	
## Zotu39:Acinetobacter_indicus	0	0	0	0	0	
## Zotu41:Lactobacillus_spc	0	1	0	0	1	
## Zotu40:Undibacterium_oligocarboniphilum	5	0	0	0	0	
##	B15-01b	B15-06	B15-07a	B15-07b	B15-11	
## Zotu19:Streptomyces_xanthophaeus	0	0	0	0	1	
## Zotu20:Pseudomonas_brenneri	9	0	0	0	1	
## Zotu28:Halomonadaceae_spc	0	0	0	2	0	
## Zotu36:Pelomonas_aquatica	4	2	3	0	1	
## Zotu17:Staphylococcus_aureus	0	0	2	1	4	
## Zotu33:Sediminibacterium_spc	13	13	2	0	3	
## Zotu42:Halomonas_desiderata	0	0	0	0	0	
## Zotu39:Acinetobacter_indicus	1	0	0	0	0	
## Zotu41:Lactobacillus_spc	0	0	0	0	0	
## Zotu40:Undibacterium_oligocarboniphilum	13	8	5	3	6	
##	B15-20	B15-24	B15-25a	B15-25b	B16-01a	
## Zotu19:Streptomyces_xanthophaeus	33	2	7	0	3	
## Zotu20:Pseudomonas_brenneri	30	2	3	0	1	
## Zotu28:Halomonadaceae_spc	10	0	1	0	0	
## Zotu36:Pelomonas_aquatica	8	0	6	0	3	
## Zotu17:Staphylococcus_aureus	8	0	2	0	0	
## Zotu33:Sediminibacterium_spc	17	0	16	1	6	
## Zotu42:Halomonas_desiderata	5	0	0	0	0	
## Zotu39:Acinetobacter_indicus	5	0	0	0	0	
## Zotu41:Lactobacillus_spc	0	1	2	1	0	
## Zotu40:Undibacterium_oligocarboniphilum	7	0	19	0	3	
##	B16-04	B16-31a	B16-31b	B20-04a	B20-08a	
## Zotu19:Streptomyces_xanthophaeus	0	13	9	2	0	

## Zotu20:Pseudomonas_brenneri	0	0	1	1	0
## Zotu28:Halomonadaceae_spc	0	4	1	0	0
## Zotu36:Pelomonas_aquatica	0	1	1	3	0
## Zotu17:Staphylococcus_aureus	0	0	3	0	0
## Zotu33:Sediminibacterium_spc	0	0	7	1	0
## Zotu42:Halomonas_desiderata	0	0	0	0	0
## Zotu39:Acinetobacter_indicus	0	0	2	0	0
## Zotu41:Lactobacillus_spc	0	1	1	0	0
## Zotu40:Undibacterium_oligocarboniphilum	0	4	1	0	1
##	B20-08b	B20-11	B23-21	B23-23a	B24-02a
## Zotu19:Streptomyces_xanthophaeus	0	0	1	0	0
## Zotu20:Pseudomonas_brenneri	0	0	3	5	0
## Zotu28:Halomonadaceae_spc	0	0	0	0	0
## Zotu36:Pelomonas_aquatica	0	0	3	20	0
## Zotu17:Staphylococcus_aureus	0	0	1	11	0
## Zotu33:Sediminibacterium_spc	0	0	0	0	0
## Zotu42:Halomonas_desiderata	0	0	0	0	0
## Zotu39:Acinetobacter_indicus	0	0	0	0	0
## Zotu41:Lactobacillus_spc	0	0	0	0	0
## Zotu40:Undibacterium_oligocarboniphilum	0	0	0	26	0
##	B24-26a	B24-26b	B24-28a	B24-29	B36-14a
## Zotu19:Streptomyces_xanthophaeus	0	1	0	0	0
## Zotu20:Pseudomonas_brenneri	0	1	0	0	0
## Zotu28:Halomonadaceae_spc	0	0	0	1	0
## Zotu36:Pelomonas_aquatica	0	1	0	1	0
## Zotu17:Staphylococcus_aureus	0	1	0	1	0
## Zotu33:Sediminibacterium_spc	0	0	2	0	1
## Zotu42:Halomonas_desiderata	0	0	0	0	0
## Zotu39:Acinetobacter_indicus	0	0	0	0	0
## Zotu41:Lactobacillus_spc	0	0	0	0	0
## Zotu40:Undibacterium_oligocarboniphilum	0	0	2	0	2
##	B36-14b	B36-30	B36-31a	B36-32	B39-15
## Zotu19:Streptomyces_xanthophaeus	15	0	1	0	0
## Zotu20:Pseudomonas_brenneri	8	1	0	0	1
## Zotu28:Halomonadaceae_spc	4	0	0	0	0
## Zotu36:Pelomonas_aquatica	2	2	0	0	4
## Zotu17:Staphylococcus_aureus	1	0	0	0	0
## Zotu33:Sediminibacterium_spc	8	8	1	0	1
## Zotu42:Halomonas_desiderata	1	0	1	0	0
## Zotu39:Acinetobacter_indicus	2	0	0	0	0
## Zotu41:Lactobacillus_spc	3	0	0	0	0
## Zotu40:Undibacterium_oligocarboniphilum	1	8	3	0	0
##	B39-16	B39-17a	B46-03	B7-07	B7-08a
## Zotu19:Streptomyces_xanthophaeus	19	0	0	9	0
## Zotu20:Pseudomonas_brenneri	6	2	0	3	0
## Zotu28:Halomonadaceae_spc	5	0	0	5	0
## Zotu36:Pelomonas_aquatica	9	0	0	4	0
## Zotu17:Staphylococcus_aureus	3	0	8	4	0
## Zotu33:Sediminibacterium_spc	19	0	0	5	0
## Zotu42:Halomonas_desiderata	1	0	0	0	0
## Zotu39:Acinetobacter_indicus	6	0	0	0	0
## Zotu41:Lactobacillus_spc	6	1	2	0	0
## Zotu40:Undibacterium_oligocarboniphilum	3	0	0	1	1
##	medium1	medium2	negative1	negative2	

## Zotu19:Streptomyces_xanthophaeus	444	859	352	532
## Zotu20:Pseudomonas_brenneri	426	883	348	524
## Zotu28:Halomonadaceae_spc	185	410	194	219
## Zotu36:Pelomonas_aquatica	113	255	120	130
## Zotu17:Staphylococcus_aureus	129	193	81	120
## Zotu33:Sediminibacterium_spc	73	234	67	83
## Zotu42:Halomonas_desiderata	60	147	78	116
## Zotu39:Acinetobacter_indicus	86	107	68	136
## Zotu41:Lactobacillus_spc	50	146	60	94
## Zotu40:Undibacterium_oligocarboniphilum	57	143	35	76
##	standard1	standard2		
## Zotu19:Streptomyces_xanthophaeus	0	0		
## Zotu20:Pseudomonas_brenneri	0	0		
## Zotu28:Halomonadaceae_spc	1	0		
## Zotu36:Pelomonas_aquatica	1	0		
## Zotu17:Staphylococcus_aureus	1104	962		
## Zotu33:Sediminibacterium_spc	0	0		
## Zotu42:Halomonas_desiderata	0	0		
## Zotu39:Acinetobacter_indicus	0	0		
## Zotu41:Lactobacillus_spc	0	0		
## Zotu40:Undibacterium_oligocarboniphilum	0	0		

only very low read numbers of most abundant ZOTUS in our samples we let ZOTUS from neg. controls in the samples and choose they way of low abundance filtering/decontam!

check for ten most abundant ZOTUs in pos.controls

```
pos.controls<-subset_samples(dataset.16S.ordi, Treatment=="standard")
sample_names(pos.controls)
```

```
## [1] "standard1" "standard2"
```

```
filtaxa <- names (sort(rowSums(otu_table(pos.controls)) ,decreasing=T))[1:10]
tax_table(dataset.16S.ordi)[filtaxa]
```

```
## Taxonomy Table:      [10 taxa by 8 taxonomic ranks]:
##
##      Domain      Phylum
## Zotu15:Salmonella_enterica      "Bacteria" "Proteobacteria"
## Zotu14:Escherichia_coli      "Bacteria" "Proteobacteria"
## Zotu18:Enterococcus_faecalis      "Bacteria" "Firmicutes"
## Zotu17:Staphylococcus_aureus      "Bacteria" "Firmicutes"
## Zotu25:Bacillus_subtilis      "Bacteria" "Firmicutes"
## Zotu27:Lactobacillus_fermentum      "Bacteria" "Firmicutes"
## Zotu23:Pseudomonas_aeruginosa      "Bacteria" "Proteobacteria"
## Zotu29>Listeria_monocytogenes      "Bacteria" "Firmicutes"
## Zotu52:Enterobacteriaceae_spc      "Bacteria" "Proteobacteria"
## Zotu147:Enterobacteriaceae_spc      "Bacteria" "Proteobacteria"
##      Class      Order
## Zotu15:Salmonella_enterica      "Gammaproteobacteria" "Enterobacteriales"
## Zotu14:Escherichia_coli      "Gammaproteobacteria" "Enterobacteriales"
## Zotu18:Enterococcus_faecalis      "Bacilli" "Lactobacillales"
## Zotu17:Staphylococcus_aureus      "Bacilli" "Bacillales"
## Zotu25:Bacillus_subtilis      "Bacilli" "Bacillales"
```



```

## Zotu27:Lactobacillus_fermentum "Bacilli" "Lactobacillales"
## Zotu23:Pseudomonas_aeruginosa "Gammaproteobacteria" "Pseudomonadales"
## Zotu29:Listeria_monocytogenes "Bacilli" "Bacillales"
## Zotu52:Enterobacteriaceae_spc "Gammaproteobacteria" "Enterobacteriales"
## Zotu147:Enterobacteriaceae_spc "Gammaproteobacteria" "Enterobacteriales"
##
## Family Genus
## Zotu15:Salmonella_enterica "Enterobacteriaceae" "Salmonella"
## Zotu14:Escherichia_coli "Enterobacteriaceae" "Escherichia"
## Zotu18:Enterococcus_faecalis "Enterococcaceae" "Enterococcus"
## Zotu17:Staphylococcus_aureus "Staphylococcaceae" "Staphylococcus"
## Zotu25:Bacillus_subtilis "Bacillaceae_1" "Bacillus"
## Zotu27:Lactobacillus_fermentum "Lactobacillaceae" "Lactobacillus"
## Zotu23:Pseudomonas_aeruginosa "Pseudomonadaceae" "Pseudomonas"
## Zotu29:Listeria_monocytogenes "Listeriaceae" "Listeria"
## Zotu52:Enterobacteriaceae_spc "Enterobacteriaceae" "Enterobacteriaceae_spc"
## Zotu147:Enterobacteriaceae_spc "Enterobacteriaceae" "Enterobacteriaceae_spc"
##
## Species
## Zotu15:Salmonella_enterica "Salmonella_enterica"
## Zotu14:Escherichia_coli "Escherichia_coli"
## Zotu18:Enterococcus_faecalis "Enterococcus_faecalis"
## Zotu17:Staphylococcus_aureus "Staphylococcus_aureus"
## Zotu25:Bacillus_subtilis "Bacillus_subtilis"
## Zotu27:Lactobacillus_fermentum "Lactobacillus_fermentum"
## Zotu23:Pseudomonas_aeruginosa "Pseudomonas_aeruginosa"
## Zotu29:Listeria_monocytogenes "Listeria_monocytogenes"
## Zotu52:Enterobacteriaceae_spc "Enterobacteriaceae_spc"
## Zotu147:Enterobacteriaceae_spc "Enterobacteriaceae_spc"
##
## best_hit
## Zotu15:Salmonella_enterica "Zotu15:Salmonella_enterica"
## Zotu14:Escherichia_coli "Zotu14:Escherichia_coli"
## Zotu18:Enterococcus_faecalis "Zotu18:Enterococcus_faecalis"
## Zotu17:Staphylococcus_aureus "Zotu17:Staphylococcus_aureus"
## Zotu25:Bacillus_subtilis "Zotu25:Bacillus_subtilis"
## Zotu27:Lactobacillus_fermentum "Zotu27:Lactobacillus_fermentum"
## Zotu23:Pseudomonas_aeruginosa "Zotu23:Pseudomonas_aeruginosa"
## Zotu29:Listeria_monocytogenes "Zotu29:Listeria_monocytogenes"
## Zotu52:Enterobacteriaceae_spc "Zotu52:Enterobacteriaceae_spc"
## Zotu147:Enterobacteriaceae_spc "Zotu147:Enterobacteriaceae_spc"

```

```
round(otu_table(dataset.16S.ordi)[filtaxa], digits = 4)
```

```

## OTU Table:      [10 taxa and 57 samples]
##                taxa are rows
##
##                B0-19a B0-19b B0-22a B0-22b B0-24 B0-27 B10-01
## Zotu15:Salmonella_enterica      3      0      0      0     10      0      1
## Zotu14:Escherichia_coli         2      0      1      0      0      3      0
## Zotu18:Enterococcus_faecalis     0      0      0      0      0      0      0
## Zotu17:Staphylococcus_aureus     7      0      7      4      2      0     14
## Zotu25:Bacillus_subtilis         0      0      0      0      0      0      0
## Zotu27:Lactobacillus_fermentum   1      0      0      0      0      0      0
## Zotu23:Pseudomonas_aeruginosa    1      0      2      0      0      0      3
## Zotu29:Listeria_monocytogenes    0      0      0      0      0      0      0
## Zotu52:Enterobacteriaceae_spc    0      0      0      0      0      0      0
## Zotu147:Enterobacteriaceae_spc   0      0      0      0      0      0      0

```

##	B10-06	B10-09a	B10-09b	B10-18	B10-24	B10-25a
## Zotu15:Salmonella_enterica	1	4	1	0	2	0
## Zotu14:Escherichia_coli	0	1	0	0	0	1
## Zotu18:Enterococcus_faecalis	0	0	0	0	0	0
## Zotu17:Staphylococcus_aureus	1	0	0	1	3	10
## Zotu25:Bacillus_subtilis	0	0	0	0	2	0
## Zotu27:Lactobacillus_fermentum	0	0	0	0	0	0
## Zotu23:Pseudomonas_aeruginosa	0	0	3	0	0	1
## Zotu29>Listeria_monocytogenes	0	0	1	0	0	0
## Zotu52:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu147:Enterobacteriaceae_spc	0	0	0	0	0	0
##	B10-25b	B10-26a	B15-01a	B15-01b	B15-06	B15-07a
## Zotu15:Salmonella_enterica	0	1	0	1	1	1
## Zotu14:Escherichia_coli	0	0	0	0	0	0
## Zotu18:Enterococcus_faecalis	0	0	0	0	0	0
## Zotu17:Staphylococcus_aureus	6	3	12	0	0	2
## Zotu25:Bacillus_subtilis	0	3	0	0	0	0
## Zotu27:Lactobacillus_fermentum	0	0	0	0	0	0
## Zotu23:Pseudomonas_aeruginosa	0	0	0	0	0	0
## Zotu29>Listeria_monocytogenes	0	0	0	0	0	0
## Zotu52:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu147:Enterobacteriaceae_spc	0	0	0	0	0	0
##	B15-07b	B15-11	B15-20	B15-24	B15-25a	B15-25b
## Zotu15:Salmonella_enterica	0	1	1	0	1	1
## Zotu14:Escherichia_coli	0	1	0	0	0	0
## Zotu18:Enterococcus_faecalis	0	0	0	0	0	0
## Zotu17:Staphylococcus_aureus	1	4	8	0	2	0
## Zotu25:Bacillus_subtilis	0	0	0	0	0	0
## Zotu27:Lactobacillus_fermentum	0	0	0	0	1	0
## Zotu23:Pseudomonas_aeruginosa	0	0	3	0	0	0
## Zotu29>Listeria_monocytogenes	0	0	0	0	0	0
## Zotu52:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu147:Enterobacteriaceae_spc	0	0	0	0	0	0
##	B16-01a	B16-04	B16-31a	B16-31b	B20-04a	B20-08a
## Zotu15:Salmonella_enterica	1	0	0	0	0	0
## Zotu14:Escherichia_coli	0	0	0	0	0	0
## Zotu18:Enterococcus_faecalis	0	0	0	0	0	0
## Zotu17:Staphylococcus_aureus	0	0	0	3	0	0
## Zotu25:Bacillus_subtilis	1	0	0	0	0	0
## Zotu27:Lactobacillus_fermentum	0	0	0	0	0	0
## Zotu23:Pseudomonas_aeruginosa	1	0	0	0	0	0
## Zotu29>Listeria_monocytogenes	0	0	0	0	0	0
## Zotu52:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu147:Enterobacteriaceae_spc	0	0	0	0	0	0
##	B20-08b	B20-11	B23-21	B23-23a	B24-02a	B24-26a
## Zotu15:Salmonella_enterica	0	0	0	0	0	0
## Zotu14:Escherichia_coli	0	0	0	0	0	0
## Zotu18:Enterococcus_faecalis	0	0	0	0	0	0
## Zotu17:Staphylococcus_aureus	0	0	1	11	0	0
## Zotu25:Bacillus_subtilis	0	0	0	2	0	0
## Zotu27:Lactobacillus_fermentum	0	0	0	0	0	0
## Zotu23:Pseudomonas_aeruginosa	0	0	0	0	0	0
## Zotu29>Listeria_monocytogenes	0	0	0	0	0	0
## Zotu52:Enterobacteriaceae_spc	0	0	0	0	0	0

plot controls

prepare data for plotting

```
negs <- subset_samples(dataset.16S.ordi, Treatment == "negative" | Treatment == "medium")

pos <- subset_samples(dataset.16S.ordi, Treatment == "standard")

Bacteria_Genus.pos <- pos %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)

Bacteria_Genus.pos$Genus<-as.character(Bacteria_Genus.pos$Genus)
Bacteria_Genus.pos$Genus[Bacteria_Genus.pos$Abundance<0.01]<-"Others"
Bacteria_Genus.pos$Class<-as.character(Bacteria_Genus.pos$Class)
Bacteria_Genus.pos$Class[Bacteria_Genus.pos$Abundance<0.01]<-"Others"

Bacteria_Genus.neg <- negs %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)

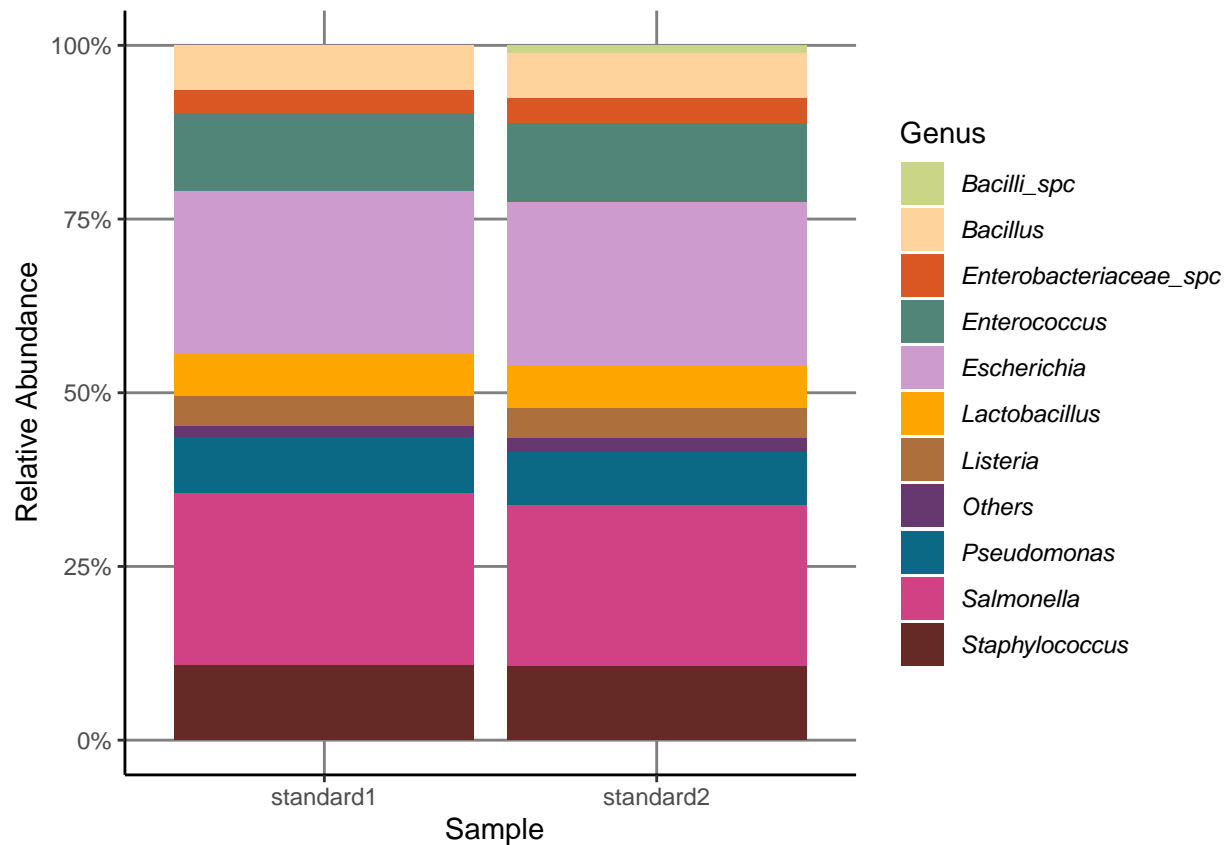
Bacteria_Genus.neg$Genus<-as.character(Bacteria_Genus.neg$Genus)
Bacteria_Genus.neg$Genus[Bacteria_Genus.neg$Abundance<0.02]<-"Others"
Bacteria_Genus.neg$Class<-as.character(Bacteria_Genus.neg$Class)
Bacteria_Genus.neg$Class[Bacteria_Genus.neg$Abundance<0.02]<-"Others"
```

plot standard controls prae decontam

```
Bacteria_Genus.pos_plot <-ggplot(Bacteria_Genus.pos, aes(x = Sample, y = Abundance, fill = Genus)) +
  scale_fill_manual(values = Family_colors, name = "Genus")

g1<-Bacteria_Genus.pos_plot +
  theme(plot.title = element_text(size = 20, face = "bold")) +
  theme(text = element_text(size=20, face = "bold"))+
  theme(axis.text.x=element_text(size = rel(0.5)))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 14))+
  theme_classic()+
  #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x=" Sample", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(legend.text = element_text(face = "italic"))

g1
```

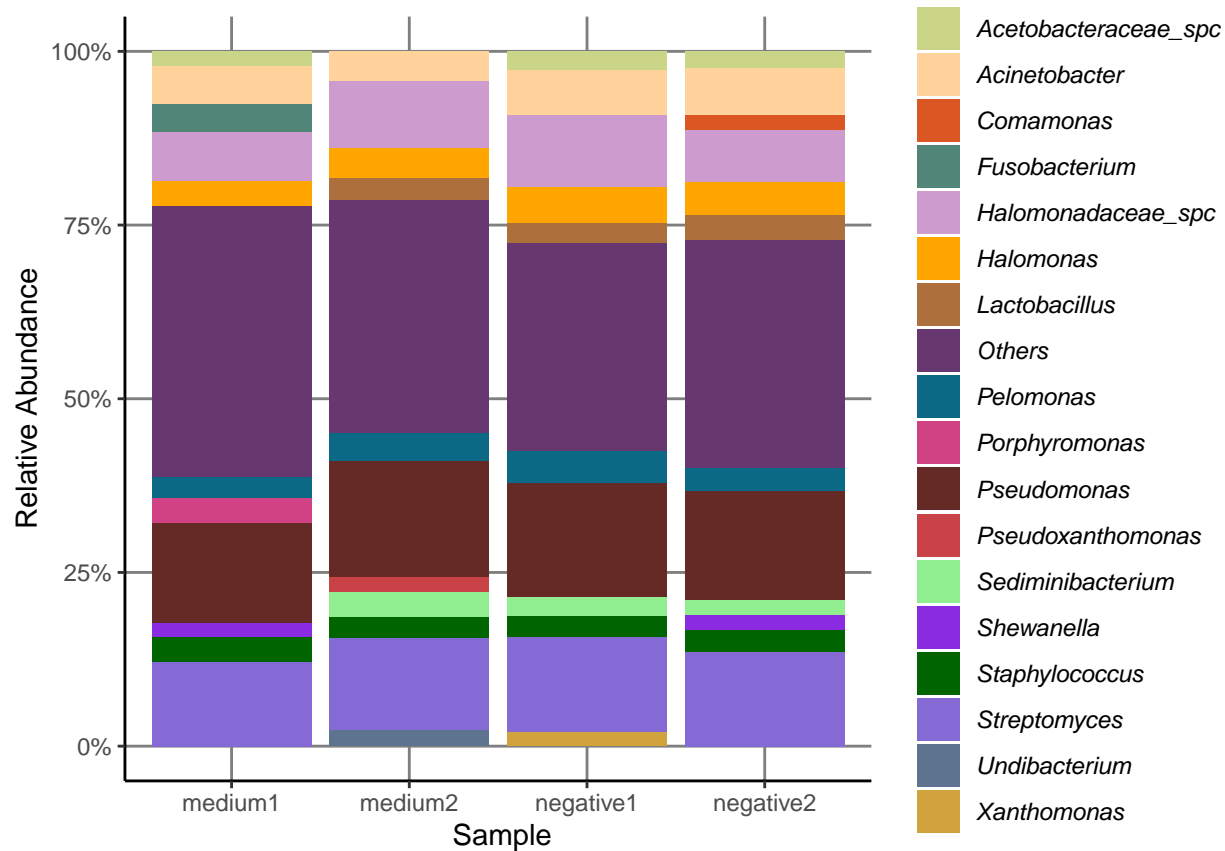


negative controls prae decontam

```
Bacteria_Genus.neg_plot <-ggplot(Bacteria_Genus.neg, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", position="fill") +
  scale_fill_manual(values = Family_colors, name = "Genus")

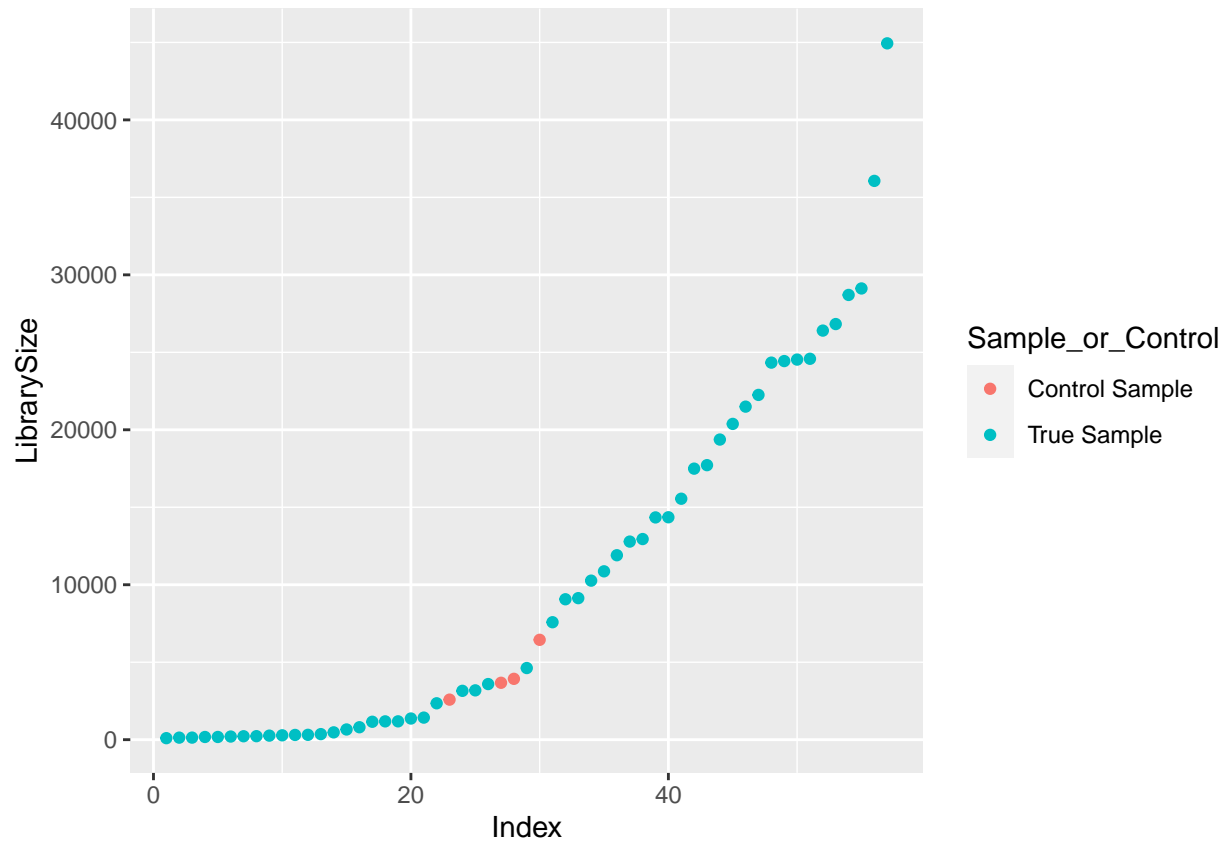
g2<-Bacteria_Genus.neg_plot +
  theme(plot.title = element_text(size = 20, face = "bold")) +
  theme(text = element_text(size=20, face = "bold"))+
  theme(axis.text.x=element_text(size = rel(0.5)))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 14))+
  theme_classic()+
  #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x=" Sample", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(legend.text = element_text(face = "italic"))

g2
```



run decontam to filter contaminating taxa

```
library(decontam)
df <- as.data.frame(sample_data(dataset.16S.ordi)) # Put sample_data into a ggplot-friendly data.frame
df$LibrarySize <- sample_sums(dataset.16S.ordi)
df <- df[order(df$LibrarySize),]
df$Index <- seq(nrow(df))
ggplot(data=df, aes(x=Index, y=LibrarySize, color=Sample_or_Control)) + geom_point()
```



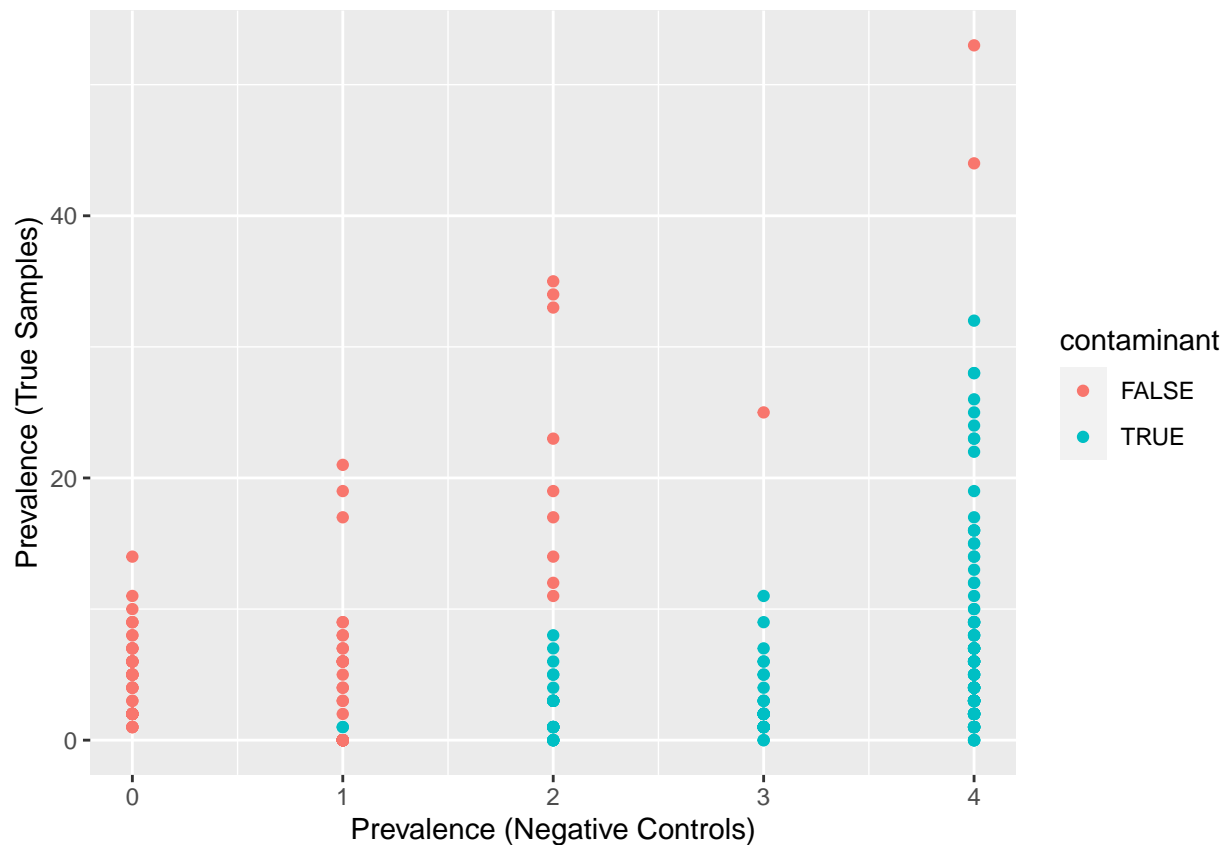
```
sample_data(dataset.16S.ordi)$is.neg <- sample_data(dataset.16S.ordi)$Sample_or_Control == "Control Sample"
contamdf.prev <- isContaminant(dataset.16S.ordi, method="prevalence", neg="is.neg")
table(contamdf.prev$contaminant)
```

```
##
## FALSE TRUE
##    114  178
```

```
ps.pa <- transform_sample_counts(dataset.16S.ordi, function(abund) 1*(abund>0))
ps.pa.neg <- prune_samples(sample_data(ps.pa)$Sample_or_Control == "Control Sample", ps.pa)
ps.pa.pos <- prune_samples(sample_data(ps.pa)$Sample_or_Control == "True Sample", ps.pa)
```

Make data.frame of prevalence in positive and negative samples

```
df.pa <- data.frame(pa.pos=taxa_sums(ps.pa.pos), pa.neg=taxa_sums(ps.pa.neg), contaminant=contamdf.prev$contaminant)
ggplot(data=df.pa, aes(x=pa.neg, y=pa.pos, color=contaminant)) + geom_point() + xlab("Prevalence (Negative)") + ylab("Prevalence (Positive)")
```



```
ps.noncontam_dataset.16S.ordi <- prune_taxa(!contamdf.prev$contaminant, dataset.16S.ordi)
ps.noncontam_dataset.16S.ordi
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 114 taxa and 57 samples ]
## sample_data() Sample Data: [ 57 samples by 17 sample variables ]
## tax_table() Taxonomy Table: [ 114 taxa by 8 taxonomic ranks ]
```

```
smin <- min(sample_sums(ps.noncontam_dataset.16S.ordi))
smean <- mean(sample_sums(ps.noncontam_dataset.16S.ordi))
smax <- max(sample_sums(ps.noncontam_dataset.16S.ordi))
cat("The minimum sample read count is:",smin)
```

```
## The minimum sample read count is: 46
```

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

```
## The average sample read count is: 9828.246
```

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

```
## The maximum sample read count is: 44924
```



```
contaminants <- subset(contamdf.prev, contaminant == "TRUE")
```

create a list of all excluded contaminants

check and plot controls again

prepare data for plotting

```
negs2 <- subset_samples(ps.noncontam_dataset.16S.ordi, Treatment == "negative" | Treatment == "medium")
negs2 <- prune_taxa(taxa_sums(negs2) > 0, negs2)
```

```
pos2 <- subset_samples(ps.noncontam_dataset.16S.ordi, Treatment == "standard")
pos2 <- prune_taxa(taxa_sums(pos2) > 0, pos2)
```

```
Bacteria_Genus.pos2 <- pos2 %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)
```

```
Bacteria_Genus.pos2$Genus<-as.character(Bacteria_Genus.pos2$Genus)
Bacteria_Genus.pos2$Genus[Bacteria_Genus.pos2$Abundance<0.01]<-"Others"
Bacteria_Genus.pos2$Class<-as.character(Bacteria_Genus.pos2$Class)
Bacteria_Genus.pos2$Class[Bacteria_Genus.pos2$Abundance<0.01]<-"Others"
```

```
Bacteria_Genus.neg2 <- negs2 %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)
```

```
Bacteria_Genus.neg2$Genus<-as.character(Bacteria_Genus.neg2$Genus)
Bacteria_Genus.neg2$Genus[Bacteria_Genus.neg2$Abundance<0.03]<-"Others"
Bacteria_Genus.neg2$Class<-as.character(Bacteria_Genus.neg2$Class)
Bacteria_Genus.neg2$Class[Bacteria_Genus.neg2$Abundance<0.03]<-"Others"
```

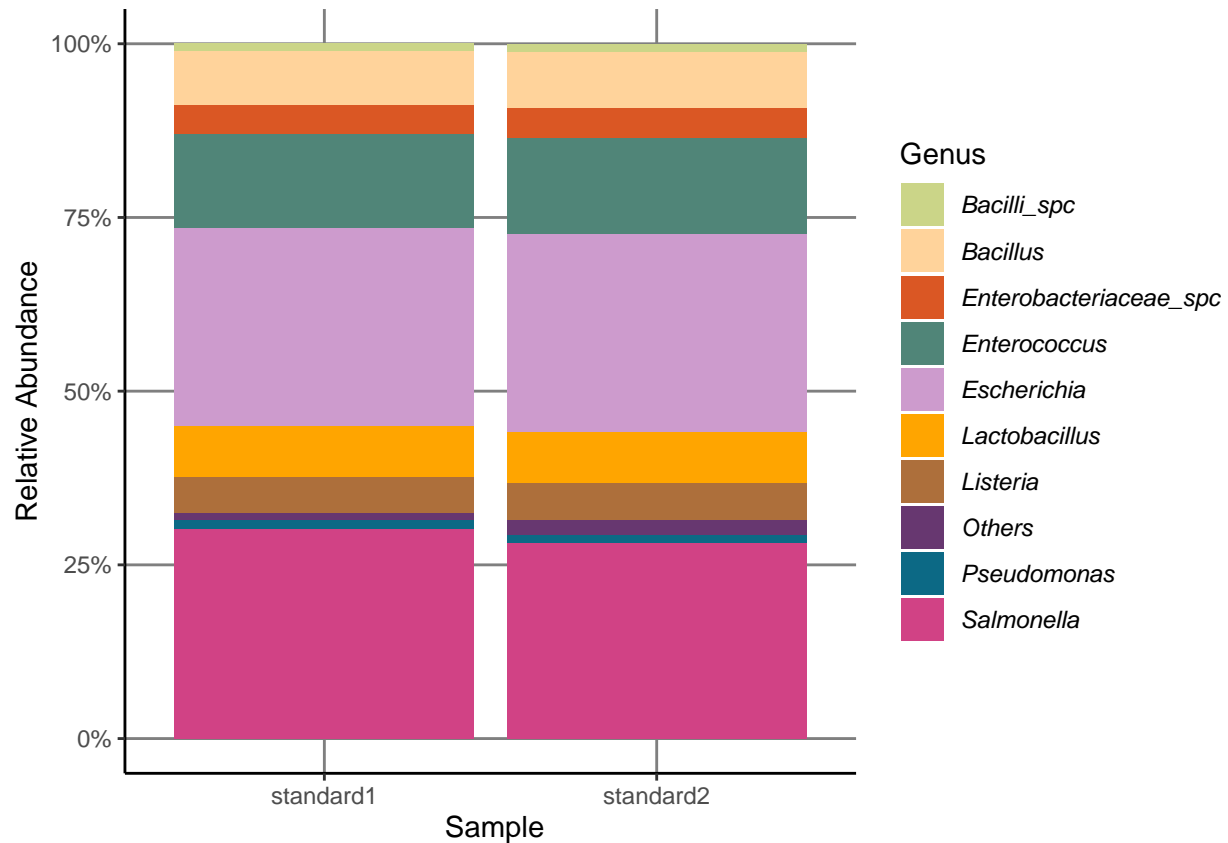
positive controls post decontam

```
Bacteria_Genus.pos2_plot <-ggplot(Bacteria_Genus.pos2, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", position="fill") +
  scale_fill_manual(values = Family_colors, name = "Genus")
```

```
g3<-Bacteria_Genus.pos2_plot +
  theme(plot.title = element_text(size = 20, face = "bold")) +
  theme(text = element_text(size=20, face = "bold"))+
  theme(axis.text.x=element_text(size = rel(0.5)))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 14))+
  theme_classic()+
  #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x=" Sample", y="Relative Abundance")+
```

```
scale_y_continuous(labels=percent_format())+
theme(legend.text = element_text(face = "italic"))
```

g3

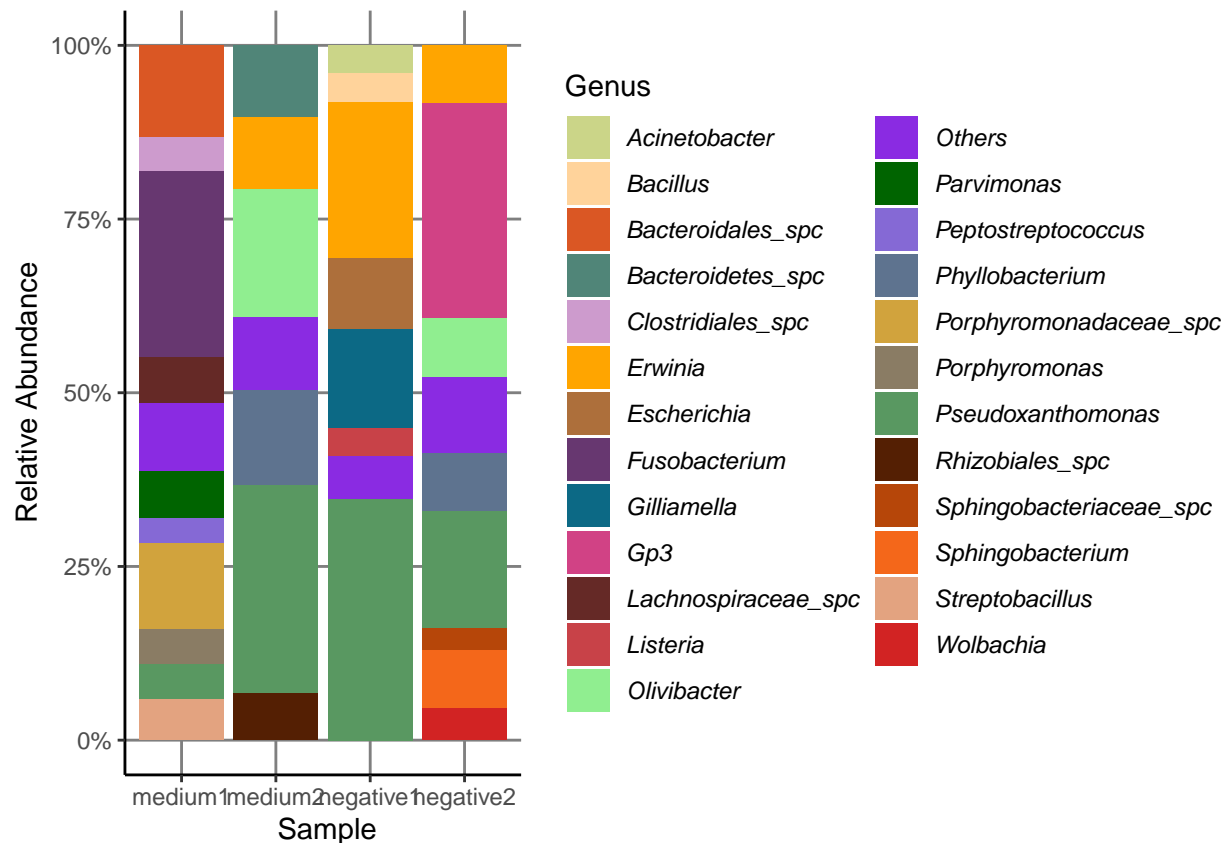


negative controls post decontam

```
Bacteria_Genus.neg2_plot <-ggplot(Bacteria_Genus.neg2, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", position="fill") +
  scale_fill_manual(values = Family_colors, name = "Genus")
```

```
g4<-Bacteria_Genus.neg2_plot +
  theme(plot.title = element_text(size = 20, face = "bold")) +
  theme(text = element_text(size=20, face = "bold"))+
  theme(axis.text.x=element_text(size = rel(0.5)))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 14))+
  theme_classic()+
  #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x=" Sample", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(legend.text = element_text(face = "italic"))
```

g4



exclude negative and positive samples for a look of the ten most abundant ZOTUS in our samples

```

bac.without.controls <- subset_samples(ps.noncontam_dataset.16S.ordi, Treatment!="medium")
bac.without.controls2 <- subset_samples(bac.without.controls, Treatment!="negative")
bwc <- subset_samples(bac.without.controls2, Treatment!="standard")

```

now have a look at the 15 most abundant ZOTUS in our samples

```

most.abundant<-subset_samples(bwc, Treatment=="control" | Treatment=="removal" | Treatment=="2nd-founda
filtaxa <- names (sort(rowSums(otu_table(most.abundant)) ,decreasing=T))[1:15]
tax_table(bwc)[filtaxa]

```

```

## Taxonomy Table:      [15 taxa by 8 taxonomic ranks]:
##
## Zotu2:Pseudoxanthomonas_spadix      "Bacteria" "Proteobacteria"
## Zotu5:Erwinia_spc                    "Bacteria" "Proteobacteria"
## Zotu8:Ochrobactrum_spc               "Bacteria" "Proteobacteria"
## Zotu7:Wolbachia_spc                  "Bacteria" "Proteobacteria"
## Zotu21:Microbacterium_pygmaeum      "Bacteria" "Actinobacteria"

```

## Zotu24:Wolbachia_spc	"Bacteria"	"Proteobacteria"
## Zotu110:Enterobacteriaceae_spc	"Bacteria"	"Proteobacteria"
## Zotu45:Pseudoxanthomonas_spc	"Bacteria"	"Proteobacteria"
## Zotu49:Microbacterium_pygmaeum	"Bacteria"	"Actinobacteria"
## Zotu9:Erwinia_spc	"Bacteria"	"Proteobacteria"
## Zotu48:Microbacterium_spc	"Bacteria"	"Actinobacteria"
## Zotu93:Stenotrophomonas_spc	"Bacteria"	"Proteobacteria"
## Zotu89:Pseudoxanthomonas_spc	"Bacteria"	"Proteobacteria"
## Zotu51:Microbacterium_spc	"Bacteria"	"Actinobacteria"
## Zotu43:Phyllobacterium_catacumbae	"Bacteria"	"Proteobacteria"
##	Class	Order
## Zotu2:Pseudoxanthomonas_spadix	"Gammaproteobacteria"	"Xanthomonadales"
## Zotu5:Erwinia_spc	"Gammaproteobacteria"	"Enterobacteriales"
## Zotu8:Ochrobactrum_spc	"Alphaproteobacteria"	"Rhizobiales"
## Zotu7:Wolbachia_spc	"Alphaproteobacteria"	"Rickettsiales"
## Zotu21:Microbacterium_pygmaeum	"Actinobacteria"	"Actinomycetales"
## Zotu24:Wolbachia_spc	"Alphaproteobacteria"	"Rickettsiales"
## Zotu110:Enterobacteriaceae_spc	"Gammaproteobacteria"	"Enterobacteriales"
## Zotu45:Pseudoxanthomonas_spc	"Gammaproteobacteria"	"Xanthomonadales"
## Zotu49:Microbacterium_pygmaeum	"Actinobacteria"	"Actinomycetales"
## Zotu9:Erwinia_spc	"Gammaproteobacteria"	"Enterobacteriales"
## Zotu48:Microbacterium_spc	"Actinobacteria"	"Actinomycetales"
## Zotu93:Stenotrophomonas_spc	"Gammaproteobacteria"	"Xanthomonadales"
## Zotu89:Pseudoxanthomonas_spc	"Gammaproteobacteria"	"Xanthomonadales"
## Zotu51:Microbacterium_spc	"Actinobacteria"	"Actinomycetales"
## Zotu43:Phyllobacterium_catacumbae	"Alphaproteobacteria"	"Rhizobiales"
##	Family	Genus
## Zotu2:Pseudoxanthomonas_spadix	"Xanthomonadaceae"	"Pseudoxanthomonas"
## Zotu5:Erwinia_spc	"Enterobacteriaceae"	"Erwinia"
## Zotu8:Ochrobactrum_spc	"Brucellaceae"	"Ochrobactrum"
## Zotu7:Wolbachia_spc	"Rickettsiaceae"	"Wolbachia"
## Zotu21:Microbacterium_pygmaeum	"Microbacteriaceae"	"Microbacterium"
## Zotu24:Wolbachia_spc	"Rickettsiaceae"	"Wolbachia"
## Zotu110:Enterobacteriaceae_spc	"Enterobacteriaceae"	"Enterobacteriaceae_spc"
## Zotu45:Pseudoxanthomonas_spc	"Xanthomonadaceae"	"Pseudoxanthomonas"
## Zotu49:Microbacterium_pygmaeum	"Microbacteriaceae"	"Microbacterium"
## Zotu9:Erwinia_spc	"Enterobacteriaceae"	"Erwinia"
## Zotu48:Microbacterium_spc	"Microbacteriaceae"	"Microbacterium"
## Zotu93:Stenotrophomonas_spc	"Xanthomonadaceae"	"Stenotrophomonas"
## Zotu89:Pseudoxanthomonas_spc	"Xanthomonadaceae"	"Pseudoxanthomonas"
## Zotu51:Microbacterium_spc	"Microbacteriaceae"	"Microbacterium"
## Zotu43:Phyllobacterium_catacumbae	"Phyllobacteriaceae"	"Phyllobacterium"
##	Species	
## Zotu2:Pseudoxanthomonas_spadix	"Pseudoxanthomonas_spadix"	
## Zotu5:Erwinia_spc	"Erwinia_spc"	
## Zotu8:Ochrobactrum_spc	"Ochrobactrum_spc"	
## Zotu7:Wolbachia_spc	"Wolbachia_spc"	
## Zotu21:Microbacterium_pygmaeum	"Microbacterium_pygmaeum"	
## Zotu24:Wolbachia_spc	"Wolbachia_spc"	
## Zotu110:Enterobacteriaceae_spc	"Enterobacteriaceae_spc"	
## Zotu45:Pseudoxanthomonas_spc	"Pseudoxanthomonas_spc"	
## Zotu49:Microbacterium_pygmaeum	"Microbacterium_pygmaeum"	
## Zotu9:Erwinia_spc	"Erwinia_spc"	
## Zotu48:Microbacterium_spc	"Microbacterium_spc"	

```

## Zotu93:Stenotrophomonas_spc "Stenotrophomonas_spc"
## Zotu89:Pseudoxanthomonas_spc "Pseudoxanthomonas_spc"
## Zotu51:Microbacterium_spc "Microbacterium_spc"
## Zotu43:Phyllobacterium_catacumbae "Phyllobacterium_catacumbae"
## best_hit
## Zotu2:Pseudoxanthomonas_spadix "Zotu2:Pseudoxanthomonas_spadix"
## Zotu5:Erwinia_spc "Zotu5:Erwinia_spc"
## Zotu8:Ochrobactrum_spc "Zotu8:Ochrobactrum_spc"
## Zotu7:Wolbachia_spc "Zotu7:Wolbachia_spc"
## Zotu21:Microbacterium_pygmaeum "Zotu21:Microbacterium_pygmaeum"
## Zotu24:Wolbachia_spc "Zotu24:Wolbachia_spc"
## Zotu110:Enterobacteriaceae_spc "Zotu110:Enterobacteriaceae_spc"
## Zotu45:Pseudoxanthomonas_spc "Zotu45:Pseudoxanthomonas_spc"
## Zotu49:Microbacterium_pygmaeum "Zotu49:Microbacterium_pygmaeum"
## Zotu9:Erwinia_spc "Zotu9:Erwinia_spc"
## Zotu48:Microbacterium_spc "Zotu48:Microbacterium_spc"
## Zotu93:Stenotrophomonas_spc "Zotu93:Stenotrophomonas_spc"
## Zotu89:Pseudoxanthomonas_spc "Zotu89:Pseudoxanthomonas_spc"
## Zotu51:Microbacterium_spc "Zotu51:Microbacterium_spc"
## Zotu43:Phyllobacterium_catacumbae "Zotu43:Phyllobacterium_catacumbae"

```

```
round(otu_table(bwc)[filtaxa], digits = 4)
```

```

## OTU Table:          [15 taxa and 51 samples]
##                      taxa are rows
##
##                      B0-19a B0-19b B0-22a B0-22b B0-24 B0-27
## Zotu2:Pseudoxanthomonas_spadix 16671 17059 42 16 319 13
## Zotu5:Erwinia_spc 0 0 33 10 44 3
## Zotu8:Ochrobactrum_spc 2 1 1 0 23 0
## Zotu7:Wolbachia_spc 1 0 895 2281 679 746
## Zotu21:Microbacterium_pygmaeum 323 293 25 9 3 15
## Zotu24:Wolbachia_spc 0 22 0 0 32 0
## Zotu110:Enterobacteriaceae_spc 0 0 0 0 0 0
## Zotu45:Pseudoxanthomonas_spc 21 124 0 0 0 0
## Zotu49:Microbacterium_pygmaeum 76 46 0 0 1 0
## Zotu9:Erwinia_spc 0 0 0 0 0 0
## Zotu48:Microbacterium_spc 58 40 0 0 0 0
## Zotu93:Stenotrophomonas_spc 0 0 0 0 0 0
## Zotu89:Pseudoxanthomonas_spc 0 0 0 0 0 0
## Zotu51:Microbacterium_spc 62 40 0 0 0 0
## Zotu43:Phyllobacterium_catacumbae 1 2 1 1 23 3
##
##                      B10-01 B10-06 B10-09a B10-09b B10-18 B10-24
## Zotu2:Pseudoxanthomonas_spadix 16 35 76 29 24 35
## Zotu5:Erwinia_spc 0 4 9 2 10 0
## Zotu8:Ochrobactrum_spc 1 1 1 0 1 0
## Zotu7:Wolbachia_spc 0 2 1 1 6 0
## Zotu21:Microbacterium_pygmaeum 0 0 0 0 0 0
## Zotu24:Wolbachia_spc 222 121 2 0 100 135
## Zotu110:Enterobacteriaceae_spc 0 0 0 0 0 0
## Zotu45:Pseudoxanthomonas_spc 0 0 0 0 0 0
## Zotu49:Microbacterium_pygmaeum 0 0 0 0 0 0
## Zotu9:Erwinia_spc 1 0 0 0 0 0
## Zotu48:Microbacterium_spc 0 0 0 0 0 0
## Zotu93:Stenotrophomonas_spc 0 0 0 0 0 0

```

## Zotu89:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu51:Microbacterium_spc	0	0	0	0	0	0
## Zotu43:Phyllobacterium_catacumbae	0	0	0	0	4	0
##	B10-25a	B10-25b	B10-26a	B15-01a	B15-01b	
## Zotu2:Pseudoxanthomonas_spadix	29	28	60	181	11	
## Zotu5:Erwinia_spc	2	2	89	85	0	
## Zotu8:Ochrobactrum_spc	1	0	1	1	0	
## Zotu7:Wolbachia_spc	0	0	1	0	163	
## Zotu21:Microbacterium_pygmaeum	0	1	0	0	0	
## Zotu24:Wolbachia_spc	1	147	58	0	0	
## Zotu110:Enterobacteriaceae_spc	0	0	0	4	0	
## Zotu45:Pseudoxanthomonas_spc	0	0	1	0	0	
## Zotu49:Microbacterium_pygmaeum	0	0	0	0	0	
## Zotu9:Erwinia_spc	0	0	0	2	0	
## Zotu48:Microbacterium_spc	0	0	0	0	0	
## Zotu93:Stenotrophomonas_spc	0	0	0	0	0	
## Zotu89:Pseudoxanthomonas_spc	0	0	0	2	0	
## Zotu51:Microbacterium_spc	0	0	0	0	0	
## Zotu43:Phyllobacterium_catacumbae	0	2	1	3	0	
##	B15-06	B15-07a	B15-07b	B15-11	B15-20	B15-24
## Zotu2:Pseudoxanthomonas_spadix	55	86	22	54	282	39
## Zotu5:Erwinia_spc	14	2	2	0	2	1
## Zotu8:Ochrobactrum_spc	0	0	23	0	0	2
## Zotu7:Wolbachia_spc	3467	3018	1124	3099	4079	1105
## Zotu21:Microbacterium_pygmaeum	0	1	0	0	0	0
## Zotu24:Wolbachia_spc	0	0	0	0	0	0
## Zotu110:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu45:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu49:Microbacterium_pygmaeum	0	0	0	0	0	0
## Zotu9:Erwinia_spc	0	0	0	0	0	0
## Zotu48:Microbacterium_spc	0	0	0	0	0	0
## Zotu93:Stenotrophomonas_spc	0	0	0	0	1	0
## Zotu89:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu51:Microbacterium_spc	0	0	0	0	0	0
## Zotu43:Phyllobacterium_catacumbae	1	2	1	0	3	0
##	B15-25a	B15-25b	B16-01a	B16-04	B16-31a	
## Zotu2:Pseudoxanthomonas_spadix	120	43	24526	24421	35797	
## Zotu5:Erwinia_spc	0	11	3	4	195	
## Zotu8:Ochrobactrum_spc	0	0	18	2	0	
## Zotu7:Wolbachia_spc	3	1091	0	0	0	
## Zotu21:Microbacterium_pygmaeum	0	2	0	0	3	
## Zotu24:Wolbachia_spc	0	0	0	2	1	
## Zotu110:Enterobacteriaceae_spc	0	0	0	0	1	
## Zotu45:Pseudoxanthomonas_spc	0	0	0	0	0	
## Zotu49:Microbacterium_pygmaeum	0	0	0	0	0	
## Zotu9:Erwinia_spc	0	0	0	0	1	
## Zotu48:Microbacterium_spc	0	0	2	0	0	
## Zotu93:Stenotrophomonas_spc	0	0	0	0	1	
## Zotu89:Pseudoxanthomonas_spc	0	0	0	0	2	
## Zotu51:Microbacterium_spc	0	0	1	0	0	
## Zotu43:Phyllobacterium_catacumbae	4	0	0	0	2	
##	B16-31b	B20-04a	B20-08a	B20-08b	B20-11	B23-21
## Zotu2:Pseudoxanthomonas_spadix	20325	21259	7488	13954	7613	41
## Zotu5:Erwinia_spc	2	0	27	1	0	13

## Zotu8:Ochrobactrum_spc	3	7424	7	6	14591	10
## Zotu7:Wolbachia_spc	6	0	15	370	20	0
## Zotu21:Microbacterium_pygmaeum	0	0	0	0	0	0
## Zotu24:Wolbachia_spc	0	2	2	0	0	162
## Zotu110:Enterobacteriaceae_spc	0	0	0	0	0	0
## Zotu45:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu49:Microbacterium_pygmaeum	0	0	0	0	0	0
## Zotu9:Erwinia_spc	0	0	0	0	0	0
## Zotu48:Microbacterium_spc	0	0	0	0	0	0
## Zotu93:Stenotrophomonas_spc	0	0	0	0	0	0
## Zotu89:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu51:Microbacterium_spc	0	0	0	0	0	0
## Zotu43:Phyllobacterium_catacumbae	0	1	2	0	2	2
##	B23-23a	B24-02a	B24-26a	B24-26b	B24-28a	
## Zotu2:Pseudoxanthomonas_spadix	73	1292	28365	3492	13767	
## Zotu5:Erwinia_spc	46	19969	15147	5361	27	
## Zotu8:Ochrobactrum_spc	9	11	18	2	0	
## Zotu7:Wolbachia_spc	0	2	4	0	0	
## Zotu21:Microbacterium_pygmaeum	0	2	2	19	278	
## Zotu24:Wolbachia_spc	0	0	1	0	0	
## Zotu110:Enterobacteriaceae_spc	0	25	316	50	0	
## Zotu45:Pseudoxanthomonas_spc	0	0	0	1	20	
## Zotu49:Microbacterium_pygmaeum	0	1	0	2	64	
## Zotu9:Erwinia_spc	0	49	247	21	1	
## Zotu48:Microbacterium_spc	0	3	0	1	42	
## Zotu93:Stenotrophomonas_spc	1	34	110	43	0	
## Zotu89:Pseudoxanthomonas_spc	0	31	124	32	0	
## Zotu51:Microbacterium_spc	2	0	0	2	42	
## Zotu43:Phyllobacterium_catacumbae	34	10	26	0	31	
##	B24-29	B36-14a	B36-14b	B36-30	B36-31a	B36-32
## Zotu2:Pseudoxanthomonas_spadix	80	12863	10757	11835	23964	23573
## Zotu5:Erwinia_spc	3	11	5	5	39	3
## Zotu8:Ochrobactrum_spc	40	0	0	0	0	2
## Zotu7:Wolbachia_spc	2	0	1	1	1	0
## Zotu21:Microbacterium_pygmaeum	25	0	0	0	194	386
## Zotu24:Wolbachia_spc	2	0	0	0	0	0
## Zotu110:Enterobacteriaceae_spc	1	0	0	0	0	0
## Zotu45:Pseudoxanthomonas_spc	0	0	0	0	2	50
## Zotu49:Microbacterium_pygmaeum	0	0	0	0	64	93
## Zotu9:Erwinia_spc	0	0	0	0	0	0
## Zotu48:Microbacterium_spc	0	0	0	0	49	67
## Zotu93:Stenotrophomonas_spc	0	0	0	0	0	0
## Zotu89:Pseudoxanthomonas_spc	0	0	0	0	0	0
## Zotu51:Microbacterium_spc	0	0	0	0	30	40
## Zotu43:Phyllobacterium_catacumbae	0	22	0	3	19	1
##	B39-15	B39-16	B39-17a	B46-03	B7-07	B7-08a
## Zotu2:Pseudoxanthomonas_spadix	26359	26612	4077	14748	19215	12684
## Zotu5:Erwinia_spc	3	6	24556	18	4	7
## Zotu8:Ochrobactrum_spc	5	1	4	3	4	18
## Zotu7:Wolbachia_spc	5	28	2	1	9	1
## Zotu21:Microbacterium_pygmaeum	0	0	1	391	0	0
## Zotu24:Wolbachia_spc	1	19	0	26	77	0
## Zotu110:Enterobacteriaceae_spc	0	0	42	0	0	0
## Zotu45:Pseudoxanthomonas_spc	0	0	0	161	2	0

## Zotu49:Microbacterium_pygmaeum	0	0	0	34	0	0
## Zotu9:Erwinia_spc	0	0	56	0	0	0
## Zotu48:Microbacterium_spc	0	0	0	29	0	0
## Zotu93:Stenotrophomonas_spc	0	0	88	0	0	0
## Zotu89:Pseudoxanthomonas_spc	0	0	83	0	0	0
## Zotu51:Microbacterium_spc	0	0	0	40	0	0
## Zotu43:Phyllobacterium_catacumbae	0	1	27	2	1	2

create subset of dataset

subset dataset bwc and exclude samples with low read counts (<500 reads)

```
sample_data(bwc)$ColSums <- colSums(otu_table(bwc))
high.reads <- subset_samples(bwc, ColSums>500)
```

reduce metadata

```
sample_data(high.reads) <- sample_data(high.reads)[,c("Sample", "Nest", "Lineage", "Treatment", "age_sampling")]
```

get rid of no read taxa in dataset

```
high.reads <- prune_taxa(taxa_sums(high.reads) > 0, high.reads)
```

extract general information on data

```
microbiome::summarize_phyloseq(high.reads)
```

```
## Compositional = N02
```

```
## [1] Min. number of reads = 7932] Max. number of reads = 449243] Total number of reads = 5404224] Average number of reads = 7.246376811594210] Number of sample variables = 7
```

```
## [[1]]
## [1] "1] Min. number of reads = 793"
##
## [[2]]
## [1] "2] Max. number of reads = 44924"
##
## [[3]]
## [1] "3] Total number of reads = 540422"
##
## [[4]]
## [1] "4] Average number of reads = 15011.7222222222"
##
## [[5]]
## [1] "5] Median number of reads = 14342"
##
## [[6]]
## [1] "7] Sparsity = 0.799919484702093"
##
## [[7]]
```



```
## [1] "6] Any OTU sum to 1 or less? YES"
##
## [[8]]
## [1] "8] Number of singletons = 5"
##
## [[9]]
## [1] "9] Percent of OTUs that are singletons \n          (i.e. exactly one read detected across all sampl
##
## [[10]]
## [1] "10] Number of sample variables are: 6"
##
## [[11]]
## [1] "Sample"          "Nest"          "Linage"          "Treatment"
## [5] "age_sampling_d" "Group"
```

```
replicates <- subset.data.frame(sample_data(high.reads))
table(replicates$Treatment)
```

```
##
## 2nd-foundation      control      removal
##              8              14              14
```

Analysis

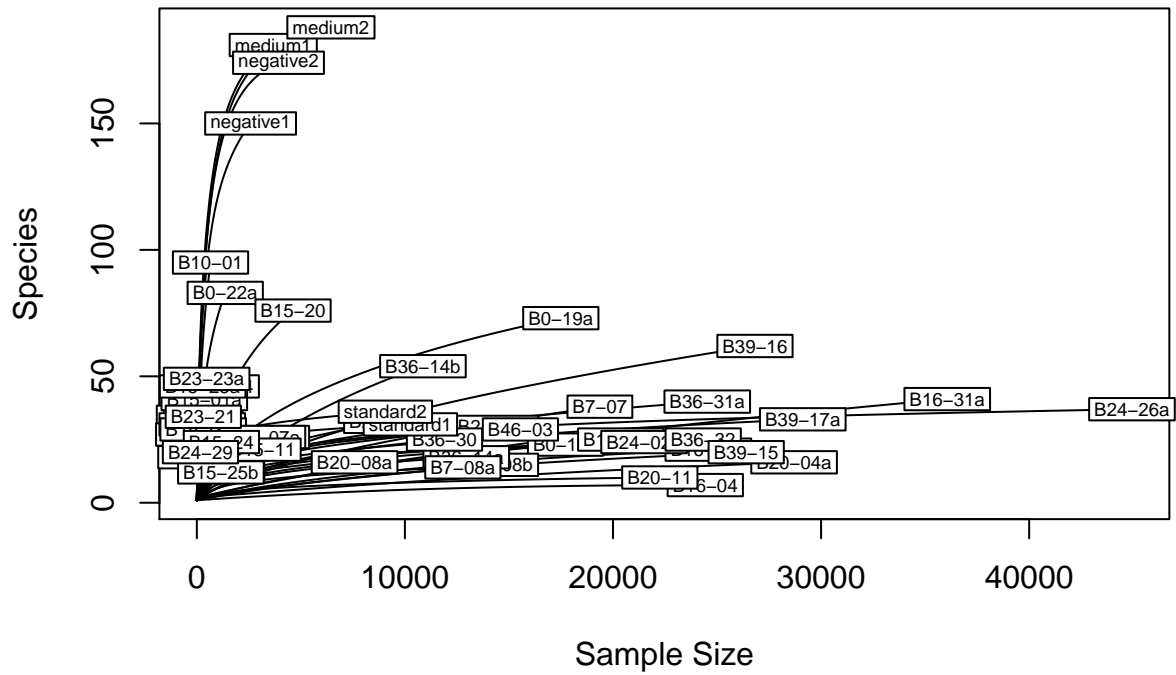
Rarefaction curves

1. all samples with controls
2. all samples with controls decontaminated
3. all samples without controls
4. all samples >500 reads

Rarefaction is used to simulate even number of reads per sample.

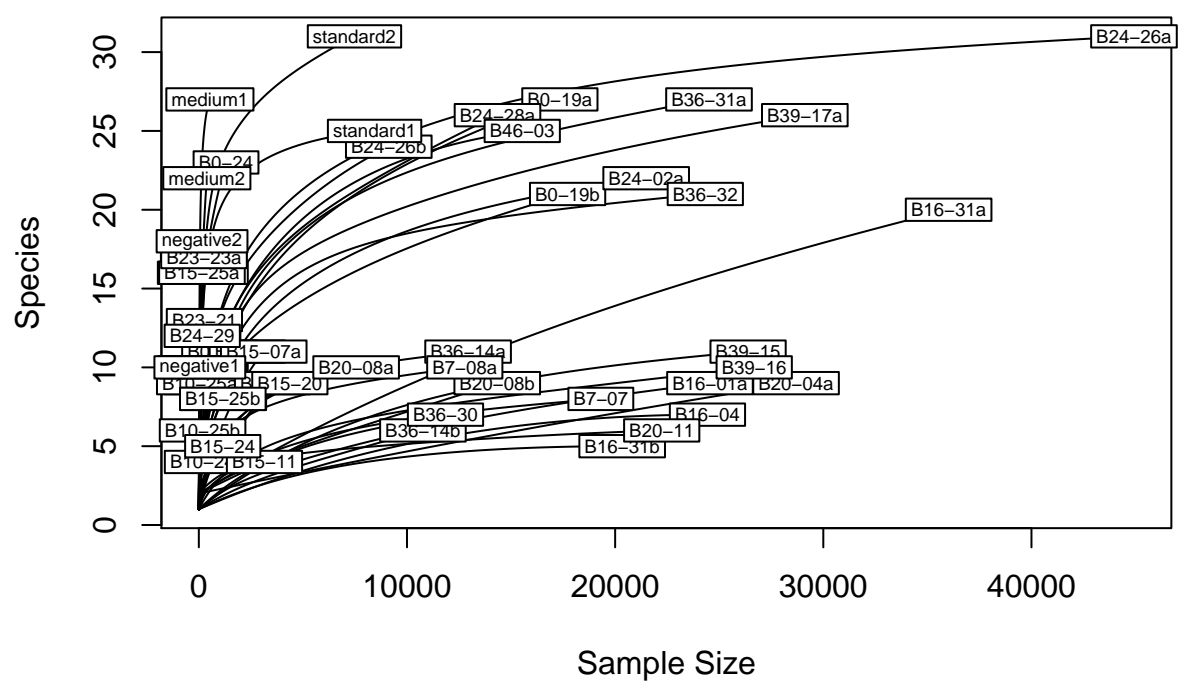
```
rarecurve(t(otu_table(dataset.16S.ordi)), cex=0.6, step = 20, main = "Rarefaction curve of all samples")
```

Rarefaction curve of all samples (incl. controls)



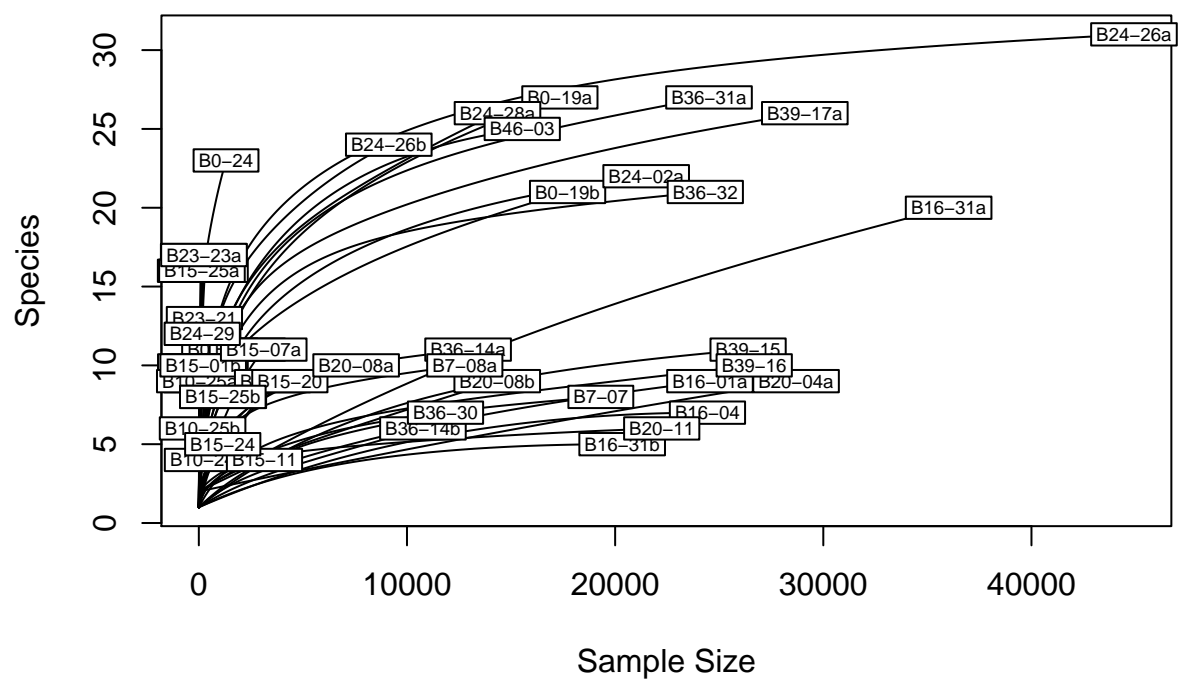
```
rarecurve(t(otu_table(ps.noncontam_dataset.16S.ordi)), cex=0.6, step = 20, main = "Rarefaction curve of
```

Rarefaction curve of all samples (incl. controls) after decontaminatic



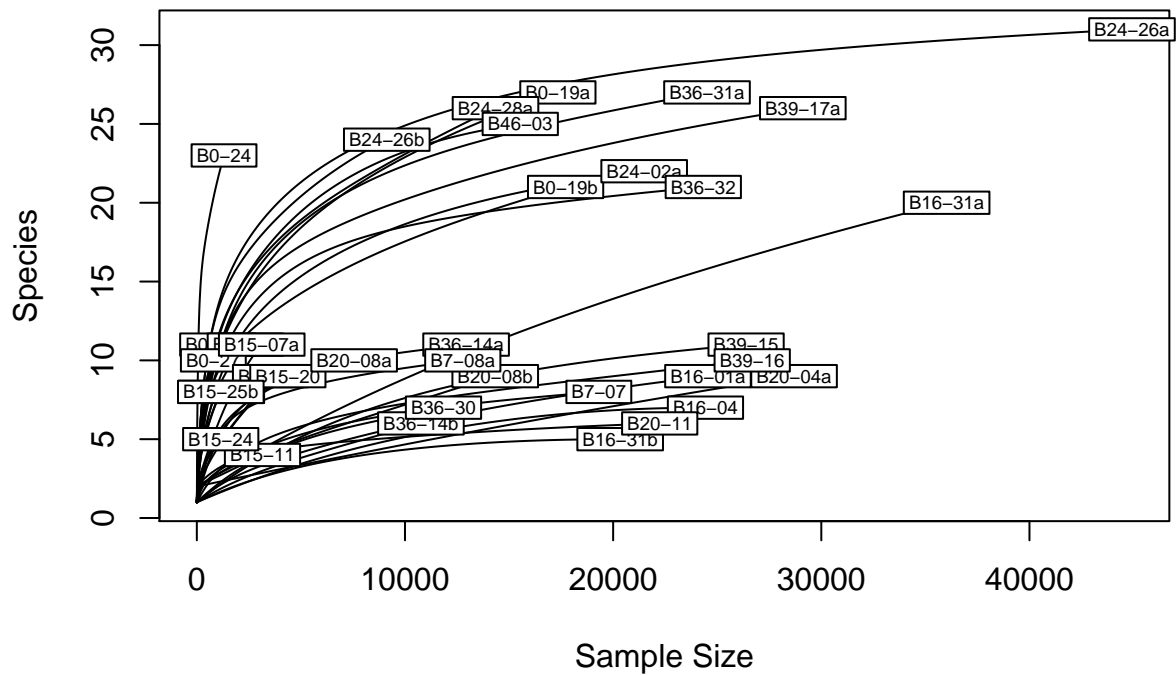
```
rarecurve(t(otu_table(bwc)), cex=0.6, step = 20, main = "Rarefaction curve of samples (excl. controls)");
```

Rarefaction curve of samples (excl. controls)



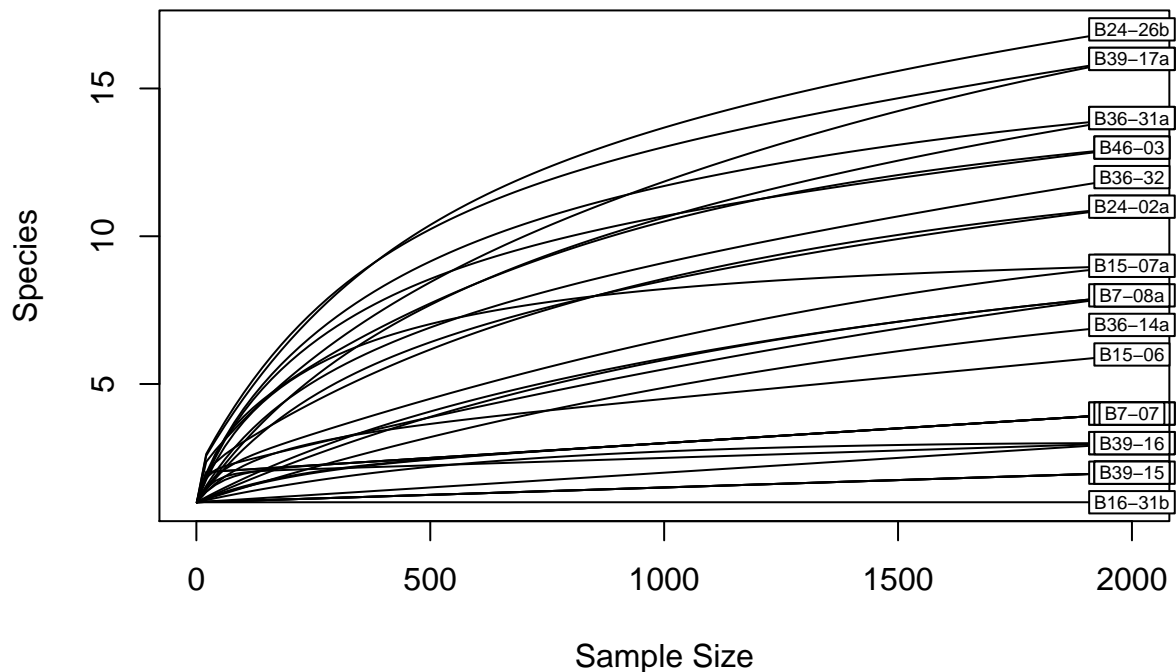
```
rarecurve(t(otu_table(high.reads)), cex=0.6, step = 20, main = "Rarefaction curve of samples >500 reads")
```

Rarefaction curve of samples >500 reads (excl. controls)



rarefy dataset to 2000 reads

```
ps2 <- high.reads
set.seed(1)
ps2 <- rarefy_even_depth(ps2, sample.size=2000, replace=FALSE, rngseed = 1)
rarecurve(t(otu_table(ps2)), cex=0.6, step = 20)
```



-> 6 samples removed because they contained fewer reads than `sample.size`. Up to first five removed samples are: B0-22a B0-24 B0-27 B15-07b B15-24 ...

15OTUs were removed because they are no longer present in any sample after random subsampling

subset data to two comparison groups

```
#control vs. removal
CR <- subset_samples(high.reads, Treatment!="2nd-foundation")
CRrf <- subset_samples(ps2, Treatment!="2nd-foundation")

#removal vs. 2nd attempt
R2nd <- subset_samples(high.reads, Treatment!="control")
R2ndrf <- subset_samples(ps2, Treatment!="control")
```

Alpha diversity

Diversity plots

This returns a table with selected diversity indicators (Shannon diversity & Observed richness).

```
tabCR <- microbiome::alpha(CRrf, index = c("diversity_shannon", "observed"))
tabR2nd <- microbiome::alpha(R2ndrf, index = c("diversity_shannon", "observed"))
```

Prepare data for visualisation Now, get the metadata (sample_data) from the phyloseq object

```
ps1.meta.CR <- meta(CRrf)
ps1.meta.R2nd <- meta(R2ndrf)
```

Add the diversity table to metadata

```
ps1.meta.CR$Shannon <- tabCR$diversity_shannon
ps1.meta.CR$Observed <- tabCR$observed

ps1.meta.R2nd$Shannon <- tabR2nd$diversity_shannon
ps1.meta.R2nd$Observed <- tabR2nd$observed
```

plot control vs. removal (rarefied, 2000 reads)

```
ps1.meta.CR$Treatment <- as.factor(ps1.meta.CR$Treatment)
ps1.meta.CR$Treatment <- factor(ps1.meta.CR$Treatment, levels = c("control", "removal"))
```

#Shannon diversity index

```
(a <- summaryBy(Shannon ~ Treatment, ps1.meta.CR, FUN = c(mean, sd)))
```

```
## Treatment Shannon.mean Shannon.sd
## 1 control 0.1539935 0.1977253
## 2 removal 0.2550152 0.2528587
```

```
p1 <- ggplot(ps1.meta.CR, aes(x = Treatment, y = Shannon))
shan_bac <- p1 + geom_jitter(position = position_jitter(w=0.2, h=0.1))+
  geom_pointrange(data = a, mapping = aes(x = Treatment, y = Shannon.mean, ymin = Shannon.mean-Shannon.sd, ymax = Shannon.mean+Shannon.sd),
  theme_bw()+
  ylim(NA, 0.95)+
  labs(x = "Treatment group", y = "Shannon Diversity Index")+
  theme(axis.title = element_text(size = 18, face = "bold"), axis.text = element_text(size = 16))
```

#Observed richness

```
(b <- summaryBy(Observed ~ Treatment, ps1.meta.CR, FUN = c(mean, sd)))
```

```
## Treatment Observed.mean Observed.sd
## 1 control 5.454545 3.933539
## 2 removal 9.769231 4.585373
```

```
p2 <- ggplot(ps1.meta.CR, aes(x = Treatment, y = Observed))
obs_bac <- p2 + geom_jitter(position = position_jitter(w=0.2, h=0.1))+
  geom_pointrange(data = b, mapping = aes(x = Treatment, y = Observed.mean, ymin = Observed.mean-Observed.sd, ymax = Observed.mean+Observed.sd),
  theme_bw()+
  ylim(NA, 17.5)+
  labs(x = "Treatment group", y = "Observed richness")+
  theme(axis.title = element_text(size = 18, face = "bold"), axis.text = element_text(size = 16))
```

removal vs. 2nd attempt (rarefied, 2000 reads)

```
ps1.meta.R2nd$Treatment <- as.factor(ps1.meta.R2nd$Treatment)
ps1.meta.R2nd$Treatment <- factor(ps1.meta.R2nd$Treatment, levels = c("removal", "2nd-foundation"), lab

#Shannon diversity index
(c <- summaryBy(Shannon ~ Treatment, ps1.meta.R2nd, FUN = c(mean, sd)))
```

```
##      Treatment Shannon.mean Shannon.sd
## 1      removal    0.2550152  0.2528587
## 2 2nd-attempt    0.2270820  0.3171305
```

```
p3 <- ggplot(ps1.meta.R2nd, aes(x = Treatment, y = Shannon))
shan_bac2<- p3 + geom_jitter(position = position_jitter(w=0.2, h=0.1))+
  geom_pointrange(data = c, mapping = aes(x = Treatment, y = Shannon.mean, ymin = Shannon.mean-Shannon
  theme_bw()+
  ylim(NA, 1)+
  labs(x = "Treatment group", y = "Shannon Diversity Index")+
  theme(axis.title = element_text(size = 18, face = "bold"), axis.text = element_text(size = 16))
```

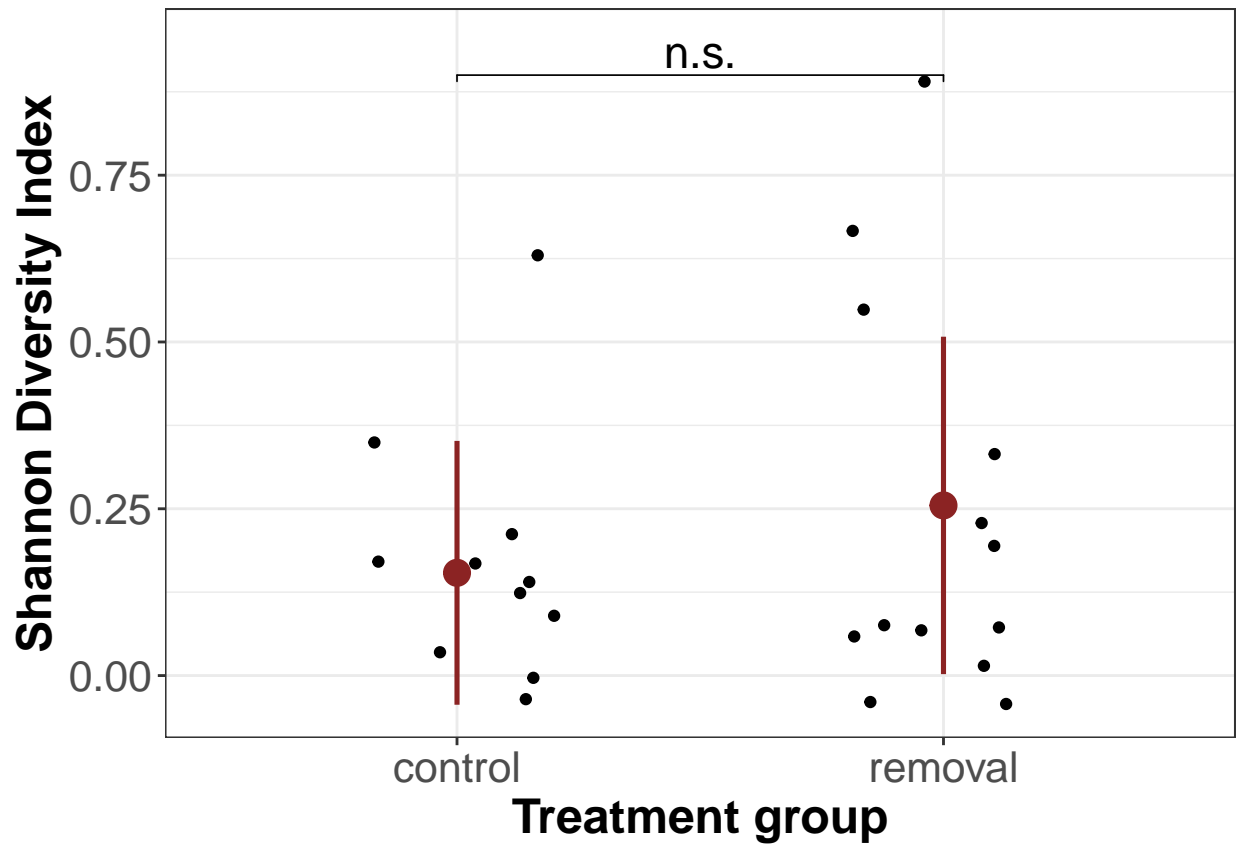
```
#Observed richness
(d <- summaryBy(Observed ~ Treatment, ps1.meta.R2nd, FUN = c(mean, sd)))
```

```
##      Treatment Observed.mean Observed.sd
## 1      removal      9.769231   4.585373
## 2 2nd-attempt      7.166667   6.274286
```

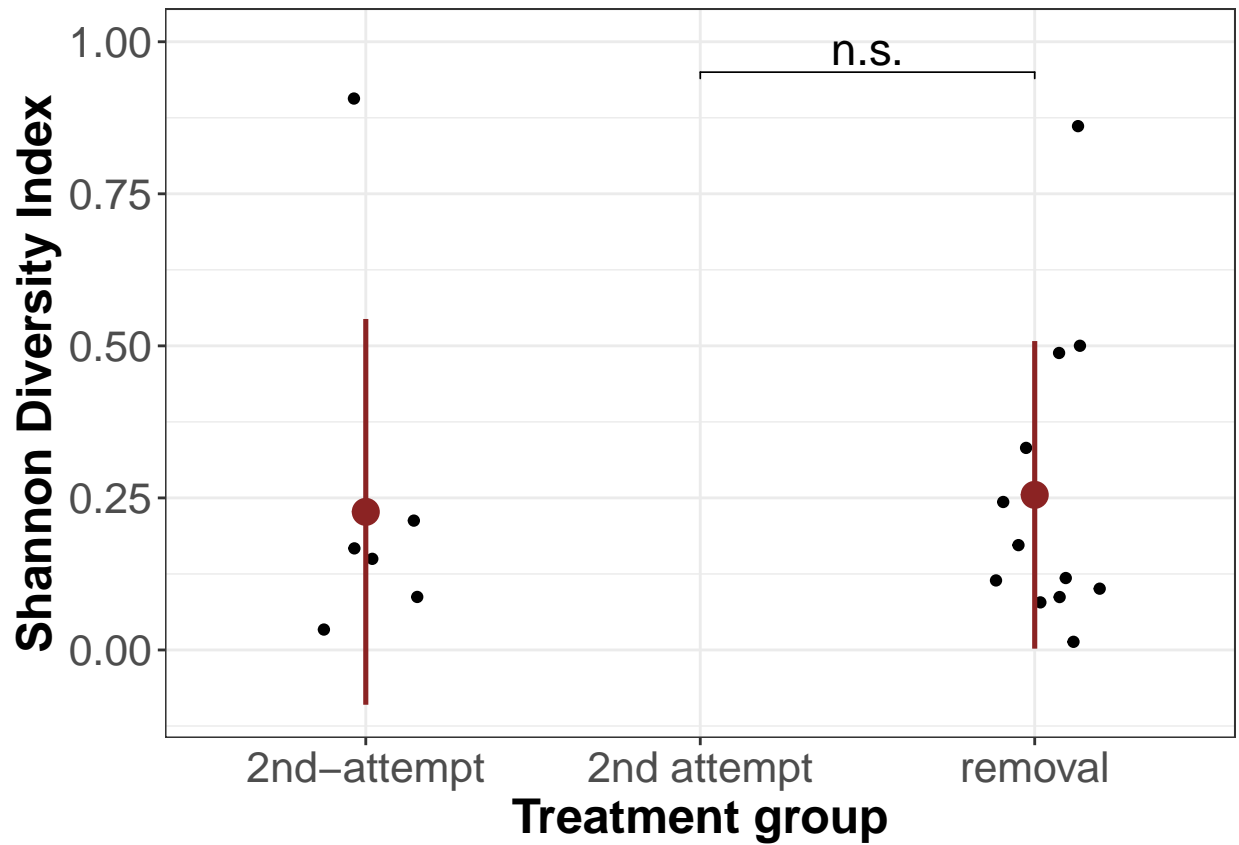
```
p4 <- ggplot(ps1.meta.R2nd, aes(x = Treatment, y = Observed))
obs_bac2 <- p4 + geom_jitter(position = position_jitter(w=0.2, h=0.1))+
  geom_pointrange(data = d, mapping = aes(x = Treatment, y = Observed.mean, ymin = Observed.mean-Obse
  theme_bw()+
  ylim(NA, 18.5)+
  labs(x = "Treatment group", y = "Observed richness")+
  theme(axis.title = element_text(size = 18, face = "bold"), axis.text = element_text(size = 16))
```

add statistical data output to plot

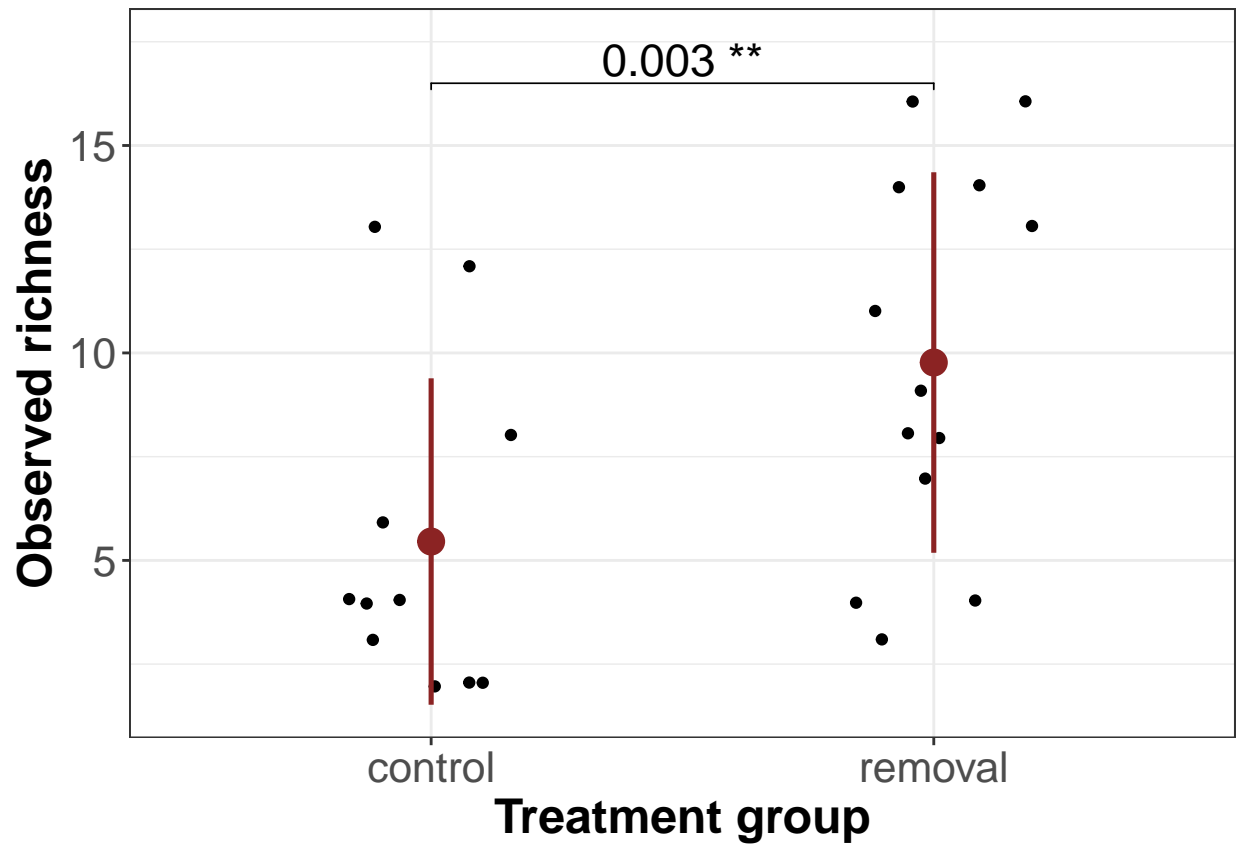
```
shan_bac <- shan_bac + font("axis.title", size = 18)+
  font("xylab", size = 18)+
  geom_bracket(
    xmin = "control", xmax = "removal", y.position = 0.9, label = "n.s.", label.size = 6,tip.le
shan_bac
```

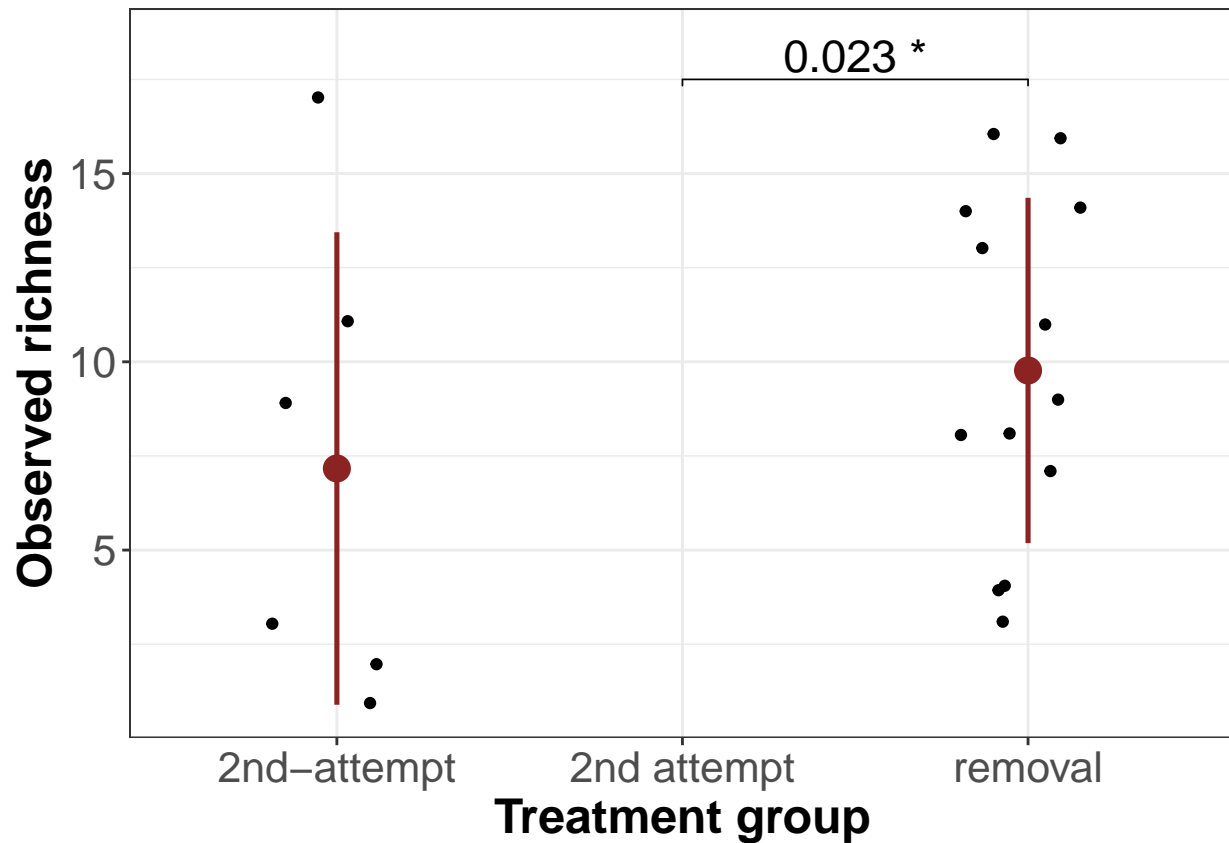
```
shan_bac2 <- shan_bac2 + font("axis.title", size = 18)+
  font("xylab", size = 18)+
  geom_bracket(
    xmin = "removal", xmax = "2nd attempt", y.position = 0.95, label = "n.s.", label.size = 6,t
  )
shan_bac2
```



```
obs_bac <- obs_bac + font("axis.title", size = 18)+
  font("xylab", size = 18)+
  geom_bracket(
    xmin = "control", xmax = "removal", y.position = 16.5, label = "0.003 **", label.size = 6,t
obs_bac
```



```
obs_bac2 <- obs_bac2 + font("axis.title", size = 18)+
  font("xylab", size = 18)+
  geom_bracket(
    xmin = "removal", xmax = "2nd attempt", y.position = 17.5, label = "0.023 **", label.size = 12)
obs_bac2
```



Statistics

Testing differences in alpha diversity

create dataframe for analysis

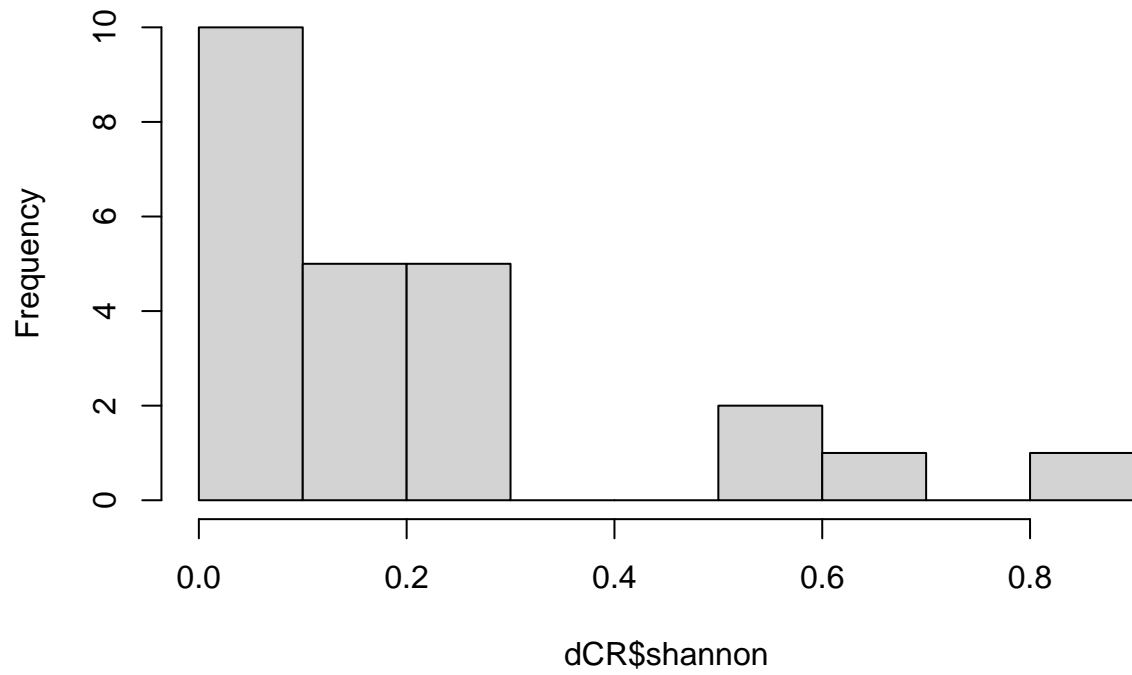
```
#control vs. removal
dCR <- meta(CRrf)
shannon <- diversity(CRrf, "shannon")
dCR$shannon <- shannon$shannon
observed <- alpha(CRrf, index = "observed", zeroes = TRUE)
dCR$observed <- observed$observed

#removal vs. 2nd attempt
dR2nd <- meta(R2ndrf)
shannon <- diversity(R2ndrf, "shannon")
dR2nd$shannon <- shannon$shannon
observed <- alpha(R2ndrf, index = "observed", zeroes = TRUE)
dR2nd$observed <- observed$observed
```

Shannon diversity test distribution

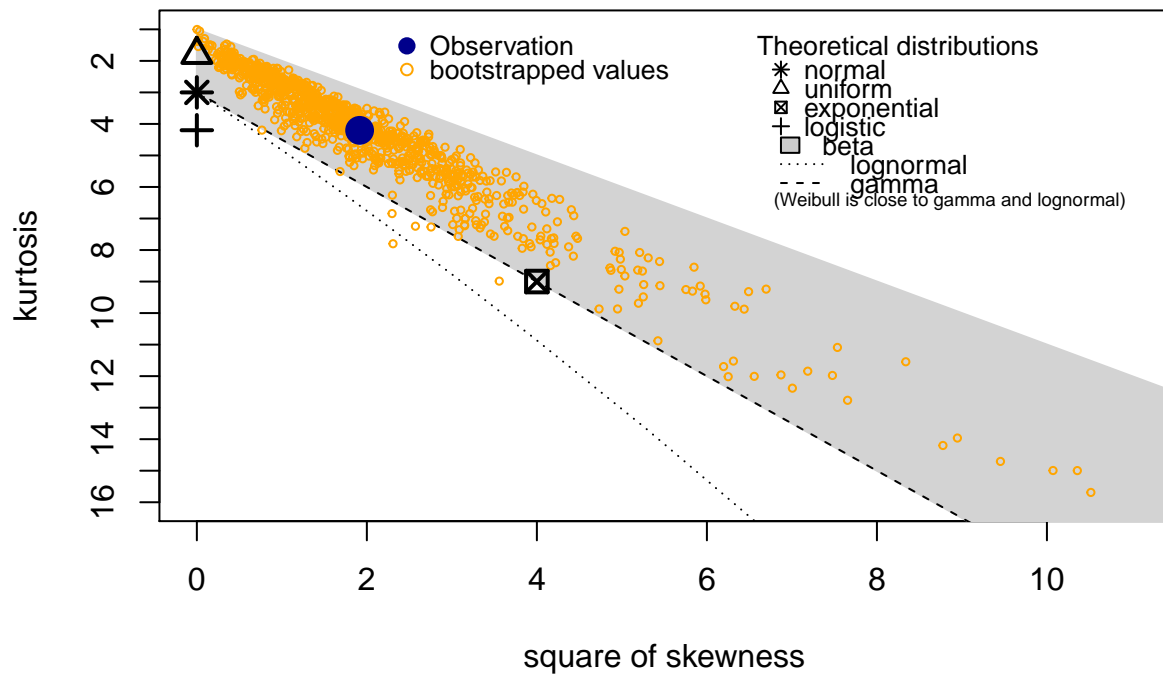
```
hist(dCR$shannon)
```

Histogram of dCR\$shannon



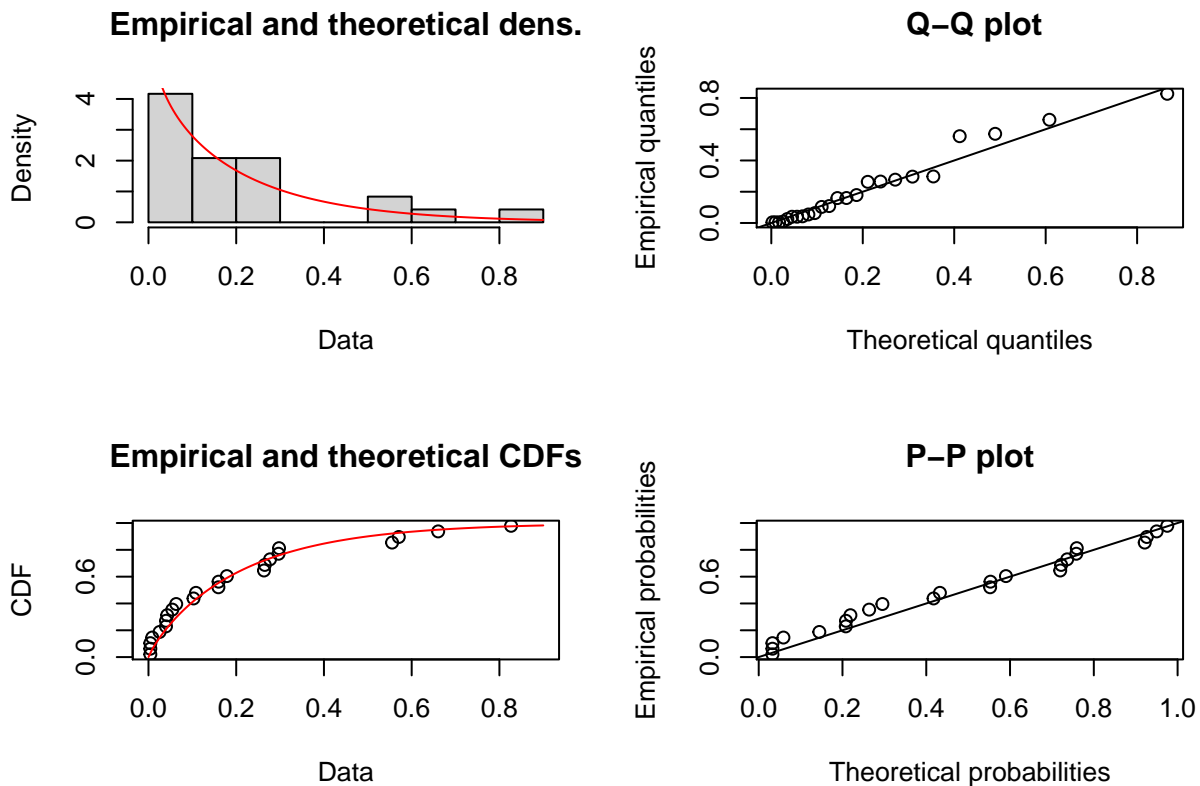
```
descdist(dCR$shannon, boot = 1000)
```

Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.004300326 max: 0.8266148
## median: 0.1338609
## mean: 0.2087136
## estimated sd: 0.2302183
## estimated skewness: 1.383827
## estimated kurtosis: 4.203764
```

```
fit.gamma <- fitdist(dCR$shannon, distr = "gamma", method = "mme")
plot(fit.gamma)
```



→ gamma distribution fits well

test with subset 1

```
dCR$Treatment <- as.factor(dCR$Treatment)
dCR <- within(dCR, Treatment <- relevel(Treatment, ref = "control"))

shan1 <- glm(shannon ~ Treatment + Linage, family = Gamma("log"), data = dCR)
Anova(shan1, type = "II")
```

```
## Analysis of Deviance Table (Type II tests)
```

```
##
```

```
## Response: shannon
```

```
##      LR Chisq Df Pr(>Chisq)
```

```
## Treatment  1.5711 1  0.21005
```

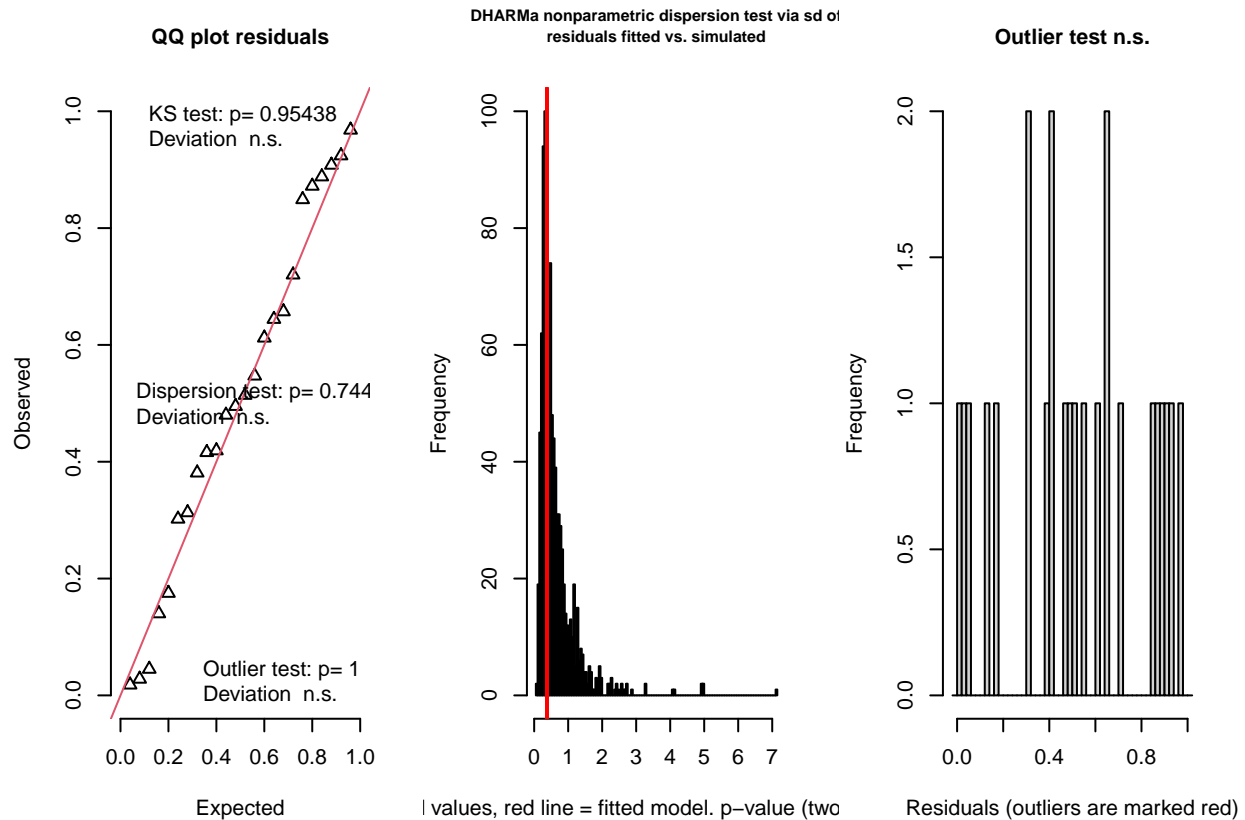
```
## Linage     19.2578 8  0.01354 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_shan1 <- simulateResiduals(shan1, n = 1000)
```

```
testResiduals(res_shan1)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.099, p-value = 0.9544
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.59005, p-value = 0.744
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
```



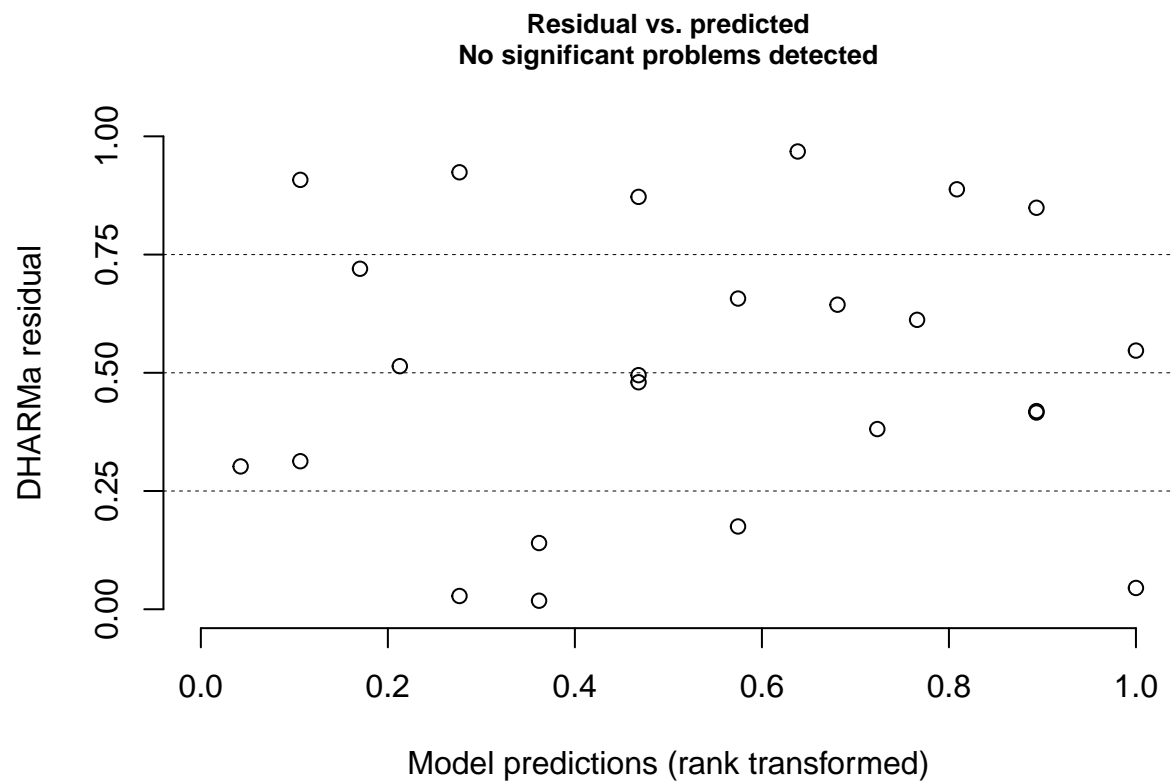
```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1424736
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.099, p-value = 0.9544
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.59005, p-value = 0.744
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1424736
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

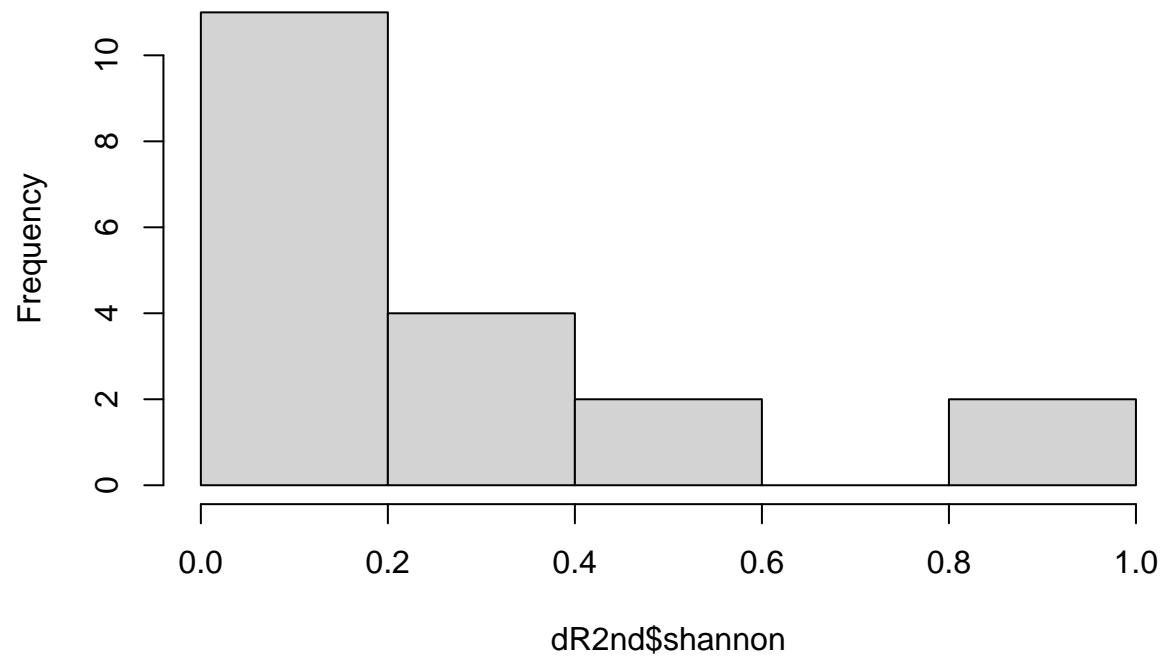
```
plotResiduals(res_shan1)
```



test with subset 2

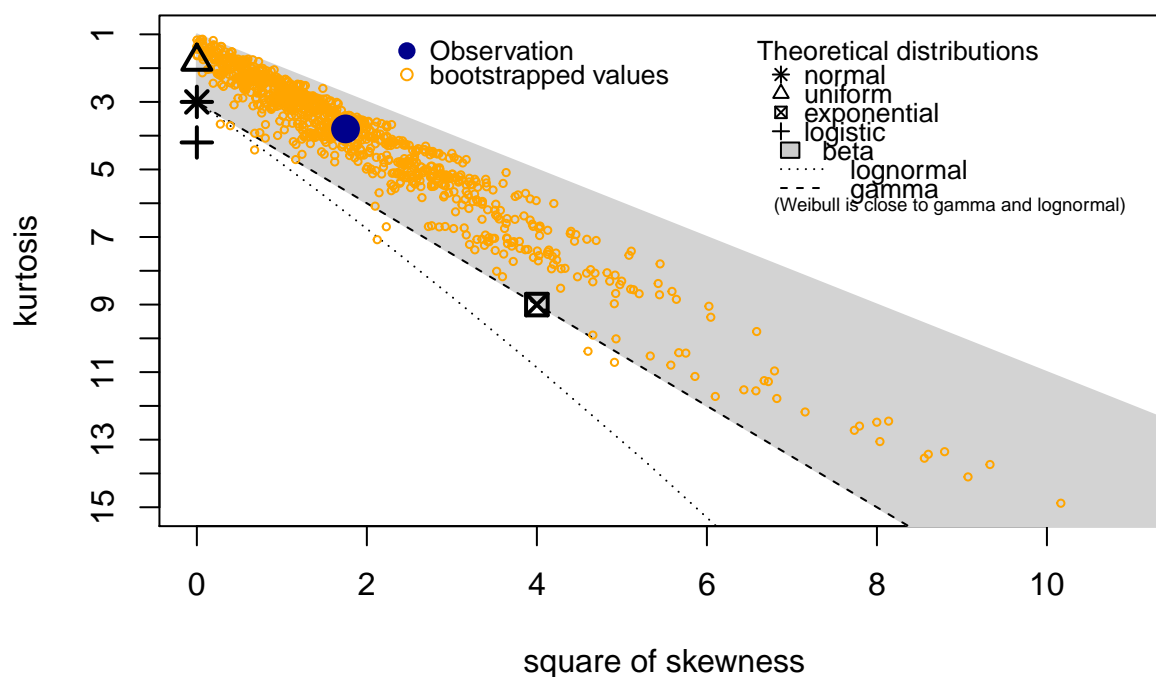
```
hist(dR2nd$shannon)
```

Histogram of dR2nd\$shannon



```
descdist(dR2nd$shannon, boot = 1000)
```

Cullen and Frey graph

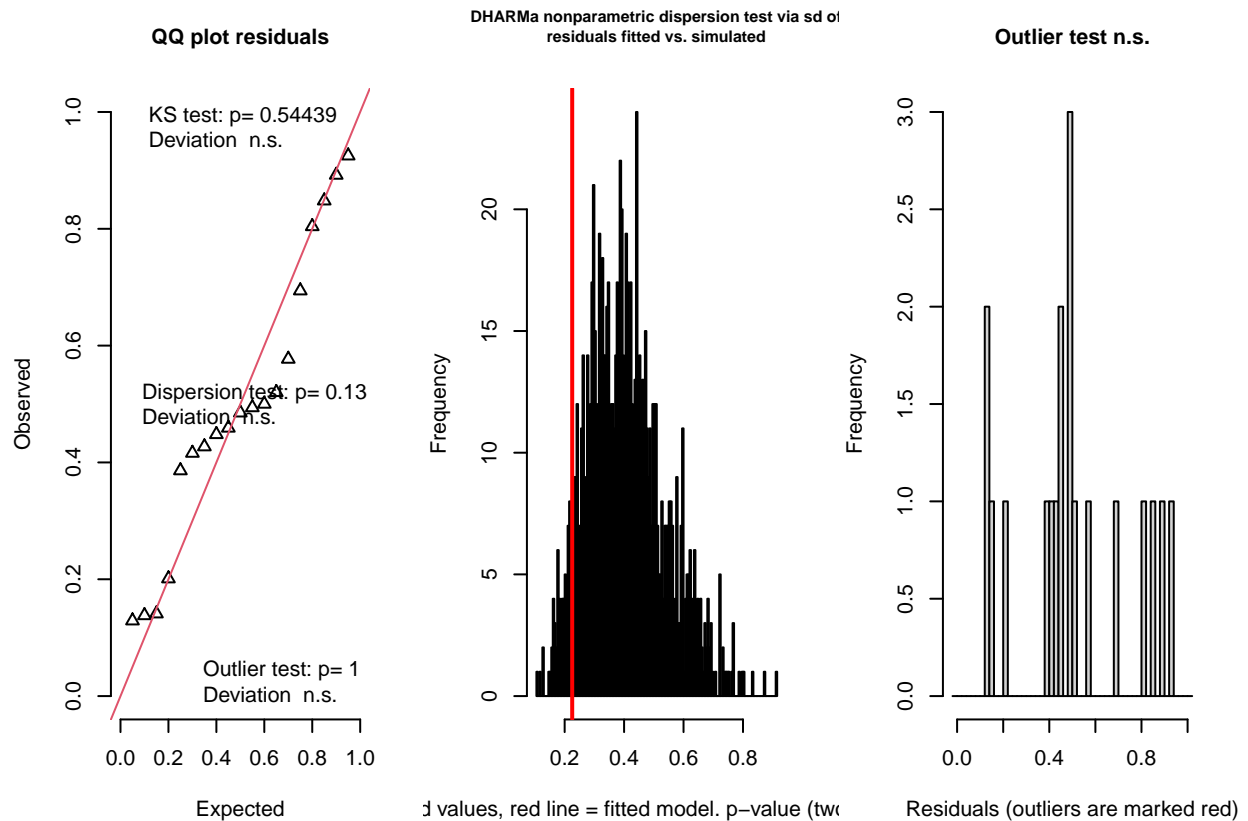


```
## summary statistics
## -----
## min: 0    max: 0.8483415
## median: 0.1594503
## mean: 0.2461942
## estimated sd: 0.2659691
## estimated skewness: 1.322517
## estimated kurtosis: 3.799409
```

```
logistic <- function(p) log(p / (1-p) +0.01)
shan1 <- lm(logistic(shannon) ~ Treatment + Linage, data = dR2nd)
Anova(shan1, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(shannon)
##          Sum Sq Df F value  Pr(>F)
## Treatment  0.622  1  0.4061 0.53829
## Linage    40.785  7  3.8062 0.02808 *
## Residuals 15.308 10
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_shan1 <- simulateResiduals(shan1, n = 1000)
testResiduals(res_shan1)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17547, p-value = 0.5444
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.55459, p-value = 0.13
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
```

```

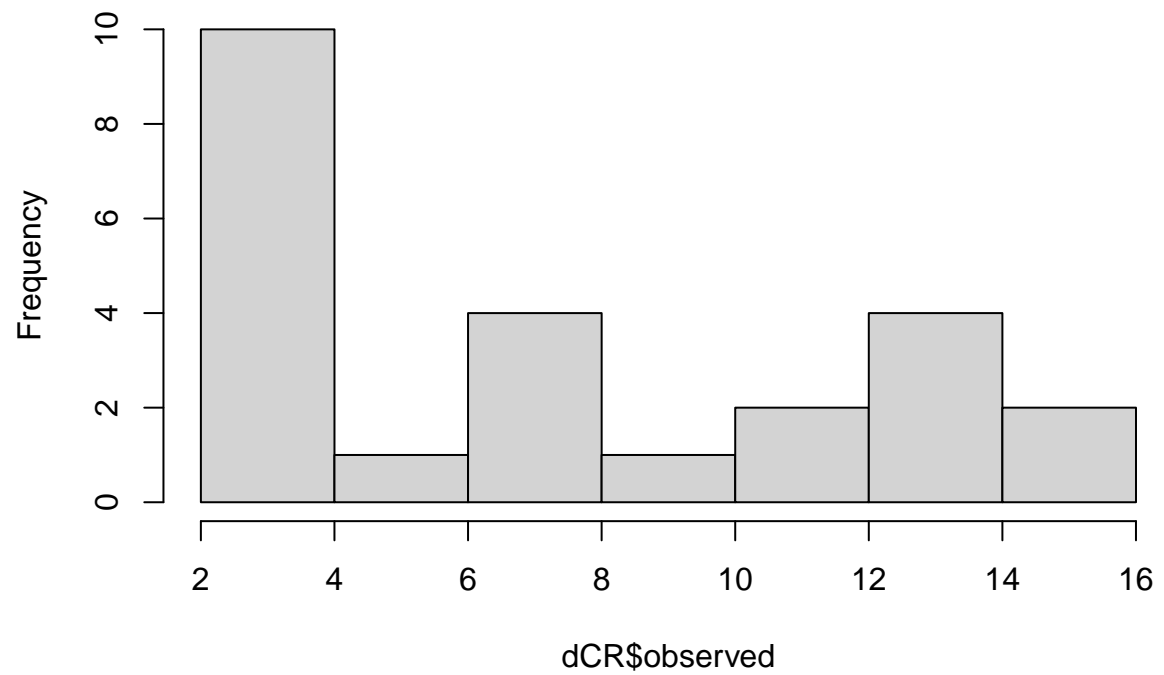
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 19, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1764669
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17547, p-value = 0.5444
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.55459, p-value = 0.13
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 19, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1764669
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

```
hist(dCR$observed)
```

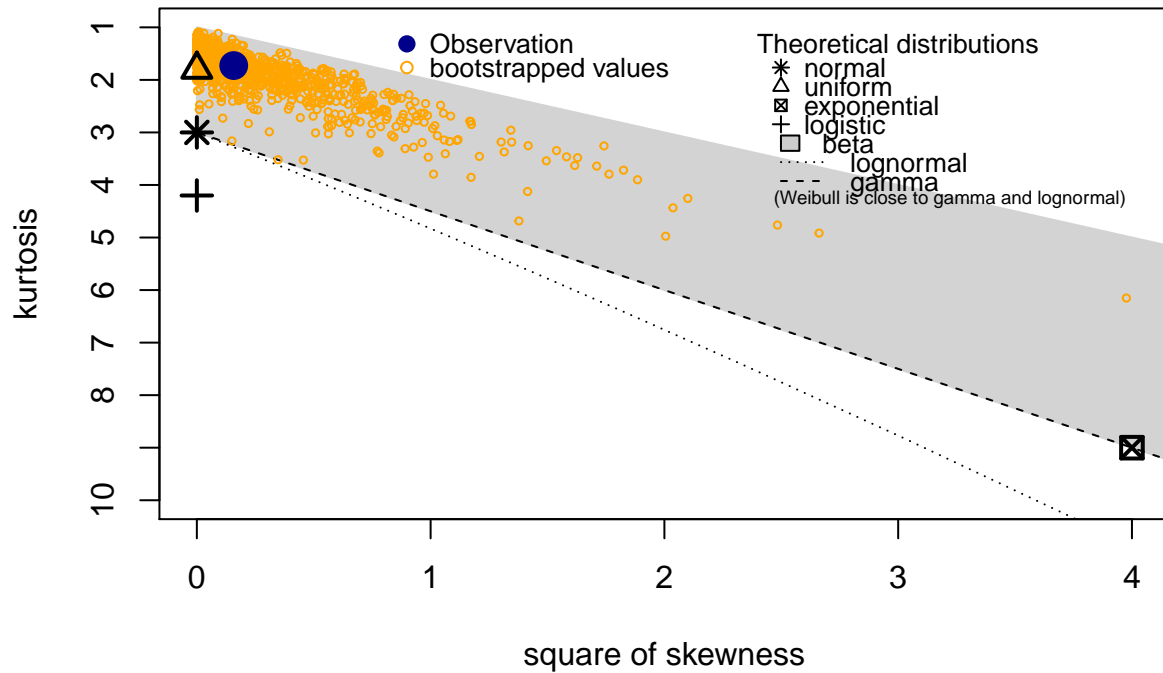
Histogram of dCR\$observed



Observed richness

```
descdist(dCR$observed, boot = 1000)
```

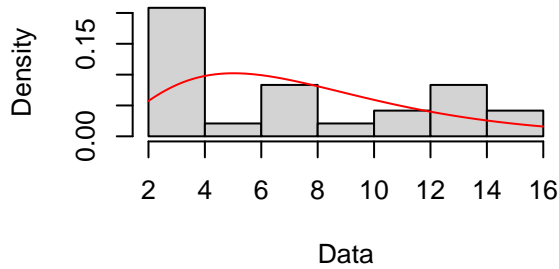
Cullen and Frey graph



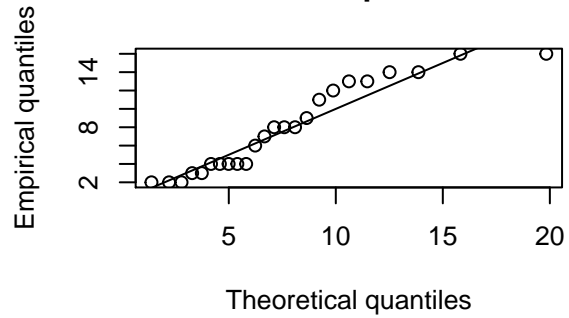
```
## summary statistics
## -----
## min: 2   max: 16
## median: 7.5
## mean: 7.791667
## estimated sd: 4.745517
## estimated skewness: 0.3971958
## estimated kurtosis: 1.7269
```

```
fit.gamma <- fitdist(dCR$observed, distr = "gamma", method = "mme")
plot(fit.gamma)
```

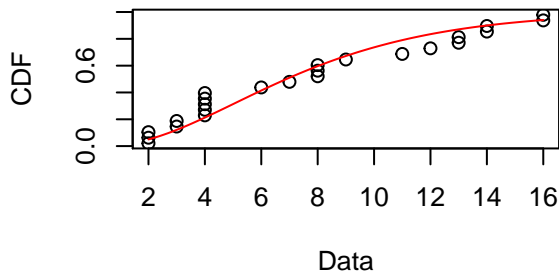

Empirical and theoretical dens.



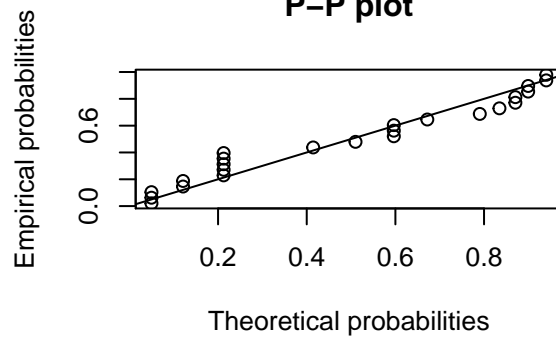
Q-Q plot



Empirical and theoretical CDFs



P-P plot



→ gamma distribution fits well

test with subset 1

```
dCR$Treatment <- as.factor(dCR$Treatment)
dCR <- within(dCR, Treatment <- relevel(Treatment, ref = "control"))

ob1 <- glm(observed ~ Treatment + Linage, family = Gamma("log"), data = dCR)
Anova(ob1, type = "II")
```

```
## Analysis of Deviance Table (Type II tests)
```

```
##
```

```
## Response: observed
```

```
##          LR Chisq Df Pr(>Chisq)
```

```
## Treatment    8.8115  1  0.002993 **
```

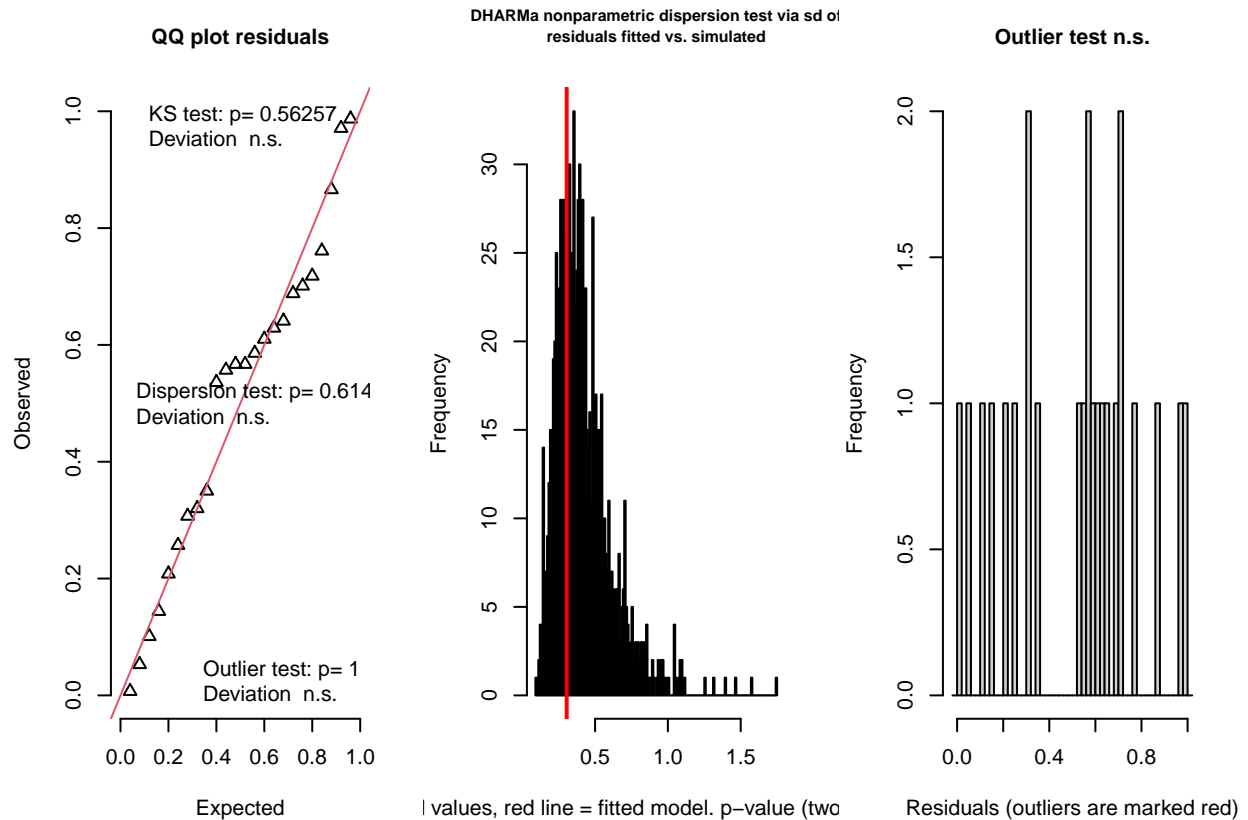
```
## Linage      19.3559  8  0.013068 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_ob1 <- simulateResiduals(ob1, n = 1000)
```

```
testResiduals(res_ob1)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.161, p-value = 0.5626
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.72767, p-value = 0.614
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
```

```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1424736
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

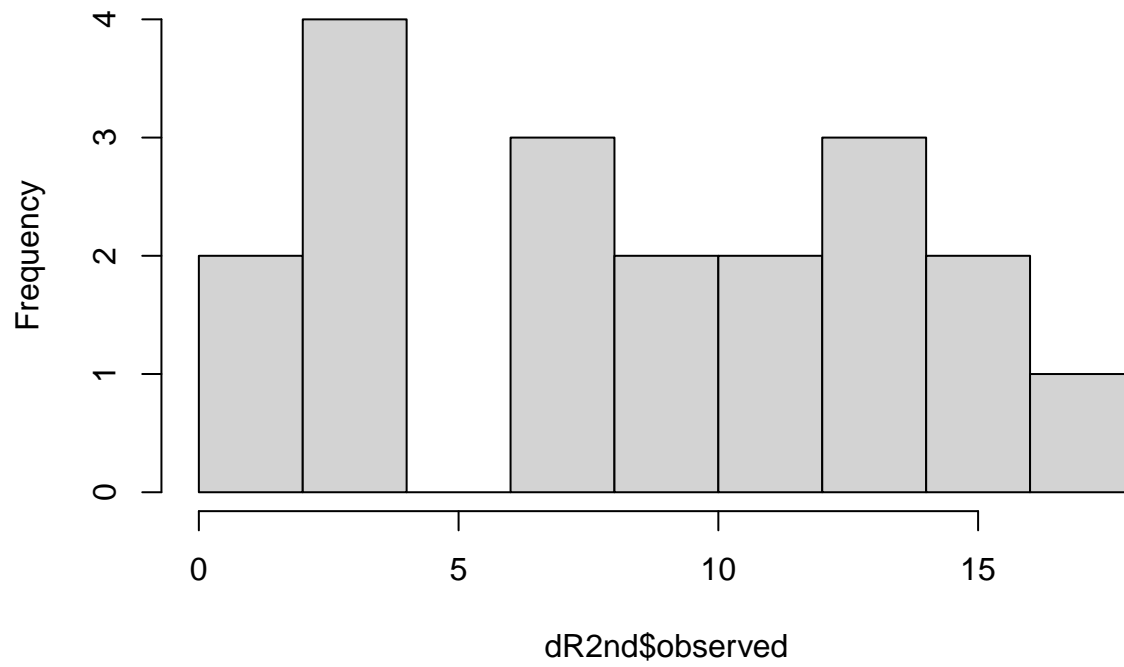
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.161, p-value = 0.5626
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.72767, p-value = 0.614
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 24, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1424736
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

test with subset 2

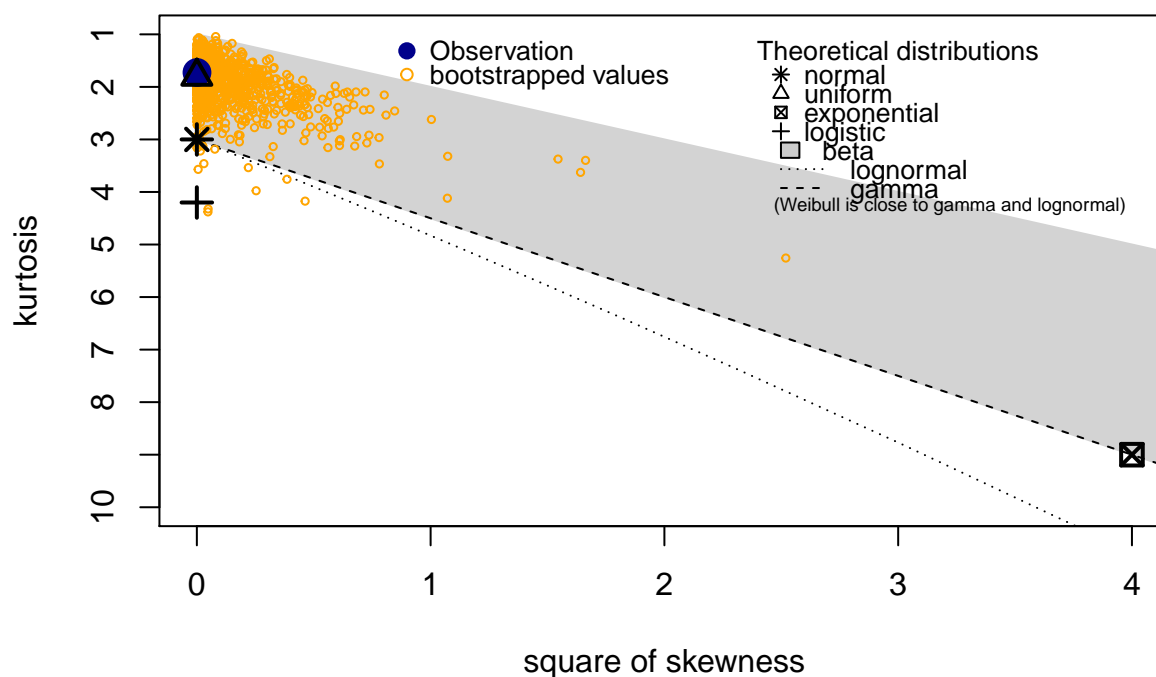
```
hist(dR2nd$observed)
```

Histogram of dR2nd\$observed



```
descdist(dR2nd$observed, boot = 1000)
```

Cullen and Frey graph

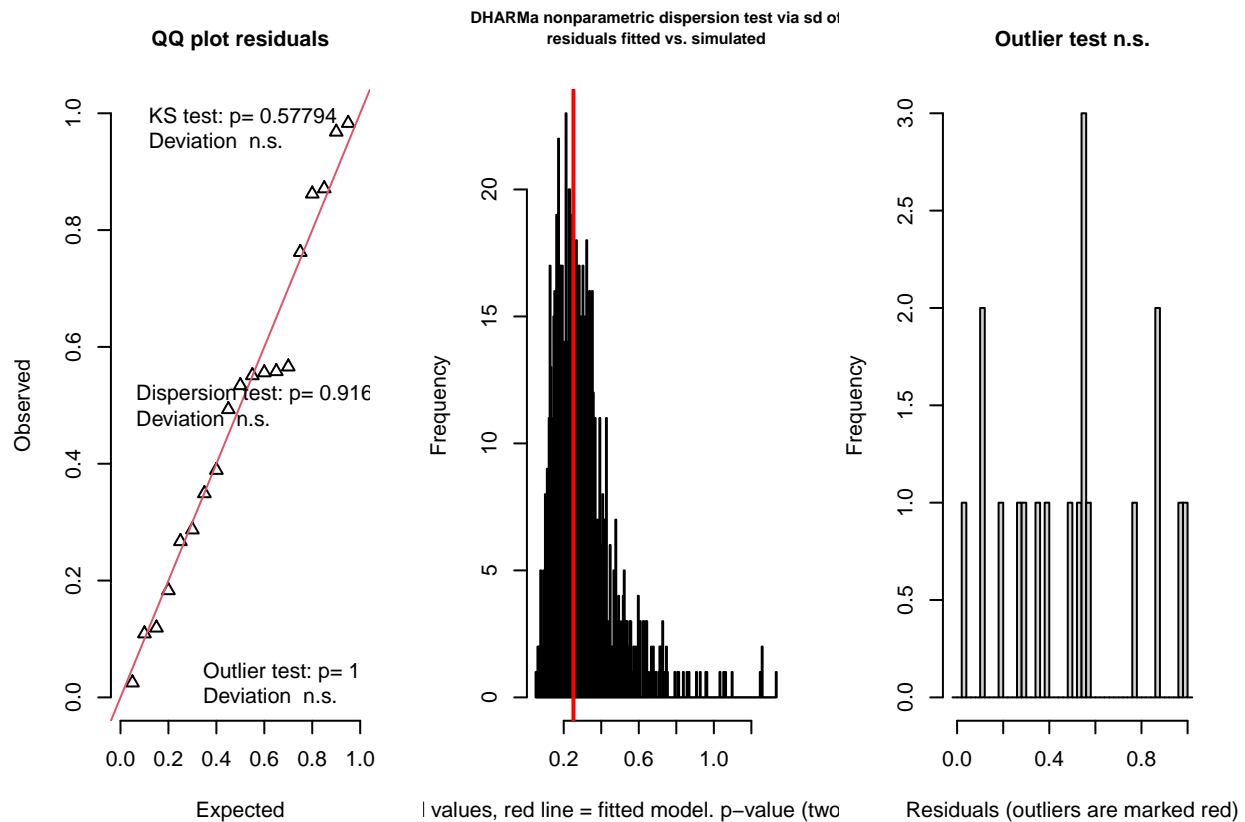


```
## summary statistics
## -----
## min: 1   max: 17
## median: 9
## mean: 8.947368
## estimated sd: 5.147531
## estimated skewness: 0.02563437
## estimated kurtosis: 1.725505
```

```
ob2 <- glm(observed ~ Treatment + Linage, data = dR2nd, family = Gamma("log"))
Anova(ob2, type = "II")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: observed
##          LR Chisq Df Pr(>Chisq)
## Treatment   5.1938  1  0.0226677 *
## Linage     29.2485  7  0.0001303 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_ob2 <- simulateResiduals(ob2, n = 1000)
testResiduals(res_ob2)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17084, p-value = 0.5779
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.84443, p-value = 0.916
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 19, p-value = 1
```

```
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1764669
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17084, p-value = 0.5779
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.84443, p-value = 0.916
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 19, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1764669
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0
```

Beta Diversity

Beta diversity and microbiome divergence

Beta diversity quantifies dissimilarity in community composition between samples. Dissimilarity can be also quantified by distance or divergence. These measures have a broad use in statistical data analysis.

transform total abundance data into compositional data

```
rel.CR <- CR %>%
  transform_sample_counts(function(x) {x/sum(x)} )

rel.R2nd <- R2nd %>%
  transform_sample_counts(function(x) {x/sum(x)} )
```

relative Abundance plots agglomerate data to Genus level, transform to rel. abundance, melt long format and sort data frame alph. by Genus

```
Bacteria_Genus1 <- CR %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)

Bacteria_Genus2 <- R2nd %>%
  tax_glom(taxrank = "Genus") %>%
  transform_sample_counts(function(x) {x/sum(x)} ) %>%
  psmelt() %>%
  arrange(Genus)
```

find the mean and standard deviation by Genus

```
rem <- subset(Bacteria_Genus1, Treatment == "removal")
mean.rem <- tapply(rem$Abundance, rem$Genus, mean)
mean.rem
```

##	Acinetobacter	Aeromicrobium	Azospirillum
##	2.290846e-05	1.988618e-05	1.420051e-04
##	Bacilli_spc	Bacillus	Bacteroidetes_spc
##	4.118107e-06	8.127397e-05	2.225556e-04
##	Chryseobacterium	Demetria	Enterobacteriaceae_spc
##	9.140731e-05	1.039717e-04	1.177764e-03
##	Enterobacteriales_spc	Erwinia	Escherichia
##	1.563085e-04	1.547910e-01	7.923877e-05
##	Flavobacterium	Gammaproteobacteria_spc	Gilliamella
##	3.340888e-05	2.453578e-06	0.000000e+00
##	Lactobacillus	Methylobacterium	Microbacteriaceae_spc
##	4.118107e-06	2.788675e-05	4.327392e-04
##	Microbacterium	Novosphingobium	Ochrobactrum
##	7.720834e-03	0.000000e+00	1.885603e-02
##	Olivibacter	Pedobacter	Phyllobacterium
##	5.318819e-04	1.765518e-04	6.294782e-04
##	Pseudomonas	Pseudoxanthomonas	Rhizobiales_spc
##	0.000000e+00	6.803950e-01	3.286906e-05
##	Roseomonas	Rubrobacter	Salmonella
##	3.412552e-05	7.504659e-06	4.062516e-05
##	Sphingobacteriaceae_spc	Sphingobacteriia_spc	Sphingobacterium
##	4.606571e-05	5.367960e-06	5.061753e-04
##	Staphylococcus	Stenotrophomonas	Veillonella
##	0.000000e+00	5.058005e-04	0.000000e+00
##	Wolbachia	Xanthomonadaceae_spc	Xanthomonadales_spc
##	1.330016e-01	1.141247e-04	0.000000e+00
##	Xanthomonas		
##	2.914381e-06		

```
SD.rem <- tapply(rem$Abundance, rem$Genus, sd)
SD.rem
```


##	Acinetobacter	Aeromicrobium	Azospirillum
##	8.571560e-05	3.922692e-05	5.313345e-04
##	Bacilli_spc	Bacillus	Bacteroidetes_spc
##	1.540855e-05	2.548339e-04	4.411361e-04
##	Chryseobacterium	Demetria	Enterobacteriaceae_spc
##	2.627674e-04	3.890266e-04	3.213563e-03
##	Enterobacteriales_spc	Erwinia	Escherichia
##	4.637591e-04	3.240921e-01	2.650834e-04
##	Flavobacterium	Gammaproteobacteria_spc	Gilliamella
##	1.100232e-04	9.180449e-06	0.000000e+00
##	Lactobacillus	Methylobacterium	Microbacteriaceae_spc
##	1.540855e-05	8.630467e-05	8.646594e-04
##	Microbacterium	Novosphingobium	Ochrobactrum
##	1.312521e-02	0.000000e+00	6.904979e-02
##	Olivibacter	Pedobacter	Phyllobacterium
##	1.043470e-03	3.478402e-04	6.520588e-04
##	Pseudomonas	Pseudoxanthomonas	Rhizobiales_spc
##	0.000000e+00	4.157842e-01	5.922467e-05
##	Roseomonas	Rubrobacter	Salmonella
##	8.435820e-05	2.140827e-05	9.302948e-05
##	Sphingobacteriaceae_spc	Sphingobacteriia_spc	Sphingobacterium
##	8.497767e-05	1.370364e-05	9.229474e-04
##	Staphylococcus	Stenotrophomonas	Veillonella
##	0.000000e+00	1.039782e-03	0.000000e+00
##	Wolbachia	Xanthomonadaceae_spc	Xanthomonadales_spc
##	3.378380e-01	3.257274e-04	0.000000e+00
##	Xanthomonas		
##	1.090462e-05		

```
sec <- subset(Bacteria_Genus2, Treatment == "2nd-foundation")
mean.sec <- tapply(sec$Abundance, sec$Genus, mean)
mean.sec
```

##	Acinetobacter	Aeromicrobium	Azospirillum
##	0.000000e+00	0.000000e+00	0.000000e+00
##	Bacilli_spc	Bacillus	Bacteroidetes_spc
##	0.000000e+00	0.000000e+00	1.085069e-04
##	Chryseobacterium	Demetria	Enterobacteriaceae_spc
##	5.367110e-05	7.055769e-06	1.082073e-03
##	Enterobacteriales_spc	Erwinia	Escherichia
##	1.506684e-04	7.573989e-02	0.000000e+00
##	Flavobacterium	Gammaproteobacteria_spc	Gilliamella
##	1.229226e-05	0.000000e+00	2.032993e-05
##	Lactobacillus	Methylobacterium	Microbacteriaceae_spc
##	2.241957e-04	0.000000e+00	1.473423e-04
##	Microbacterium	Novosphingobium	Ochrobactrum
##	4.567880e-03	1.161063e-05	2.549932e-03
##	Olivibacter	Pedobacter	Phyllobacterium
##	5.367110e-05	1.767884e-04	1.740752e-04
##	Pseudomonas	Pseudoxanthomonas	Rhizobiales_spc
##	6.639364e-04	5.493352e-01	0.000000e+00
##	Roseomonas	Rubrobacter	Salmonella
##	0.000000e+00	1.062925e-04	1.085069e-04
##	Sphingobacteriaceae_spc	Sphingobacteriia_spc	Sphingobacterium

```
##          0.000000e+00          0.000000e+00          8.719308e-06
##      Staphylococcus      Stenotrophomonas      Veillonella
##          0.000000e+00          5.889766e-04          1.610133e-04
##          Wolbachia      Xanthomonadaceae_spc      Xanthomonadales_spc
##          3.637075e-01          2.191541e-04          1.369713e-05
##          Xanthomonas
##          7.055769e-06
```

```
SD.sec <- tapply(sec$Abundance, sec$Genus, sd)
SD.sec
```

```
##          Acinetobacter          Aeromicrobium          Azospirillum
##          0.000000e+00          0.000000e+00          0.000000e+00
##          Bacilli_spc          Bacillus          Bacteroidetes_spc
##          0.000000e+00          0.000000e+00          3.069040e-04
##          Chryseobacterium          Demetria      Enterobacteriaceae_spc
##          1.518048e-04          1.995673e-05          3.060565e-03
##      Enterobacteriales_spc          Erwinia          Escherichia
##          4.261546e-04          2.077148e-01          0.000000e+00
##          Flavobacterium      Gammaproteobacteria_spc          Gilliamella
##          3.476776e-05          0.000000e+00          3.814795e-05
##          Lactobacillus          Methylobacterium      Microbacteriaceae_spc
##          5.974746e-04          0.000000e+00          3.370360e-04
##          Microbacterium          Novosphingobium          Ochrobactrum
##          9.421317e-03          3.283981e-05          6.873753e-03
##          Olivibacter          Pedobacter          Phyllobacterium
##          1.518048e-04          4.499583e-04          3.111645e-04
##          Pseudomonas          Pseudoxanthomonas          Rhizobiales_spc
##          9.962397e-04          4.818013e-01          0.000000e+00
##          Roseomonas          Rubrobacter          Salmonella
##          0.000000e+00          3.006406e-04          3.069040e-04
##      Sphingobacteriaceae_spc      Sphingobacteriia_spc      Sphingobacterium
##          0.000000e+00          0.000000e+00          2.466193e-05
##          Staphylococcus          Stenotrophomonas          Veillonella
##          0.000000e+00          1.665877e-03          4.554144e-04
##          Wolbachia      Xanthomonadaceae_spc      Xanthomonadales_spc
##          4.945458e-01          6.198613e-04          3.874133e-05
##          Xanthomonas
##          1.995673e-05
```

```
control <- subset(Bacteria_Genus1, Treatment == "control")
mean.control <- tapply(control$Abundance, control$Genus, mean)
mean.control
```

```
##          Acinetobacter          Aeromicrobium          Azospirillum
##          3.677936e-03          5.465078e-05          0.000000e+00
##          Bacilli_spc          Bacillus          Bacteroidetes_spc
##          0.000000e+00          1.074849e-04          6.296744e-04
##          Chryseobacterium          Demetria      Enterobacteriaceae_spc
##          1.034056e-04          9.290633e-04          0.000000e+00
##      Enterobacteriales_spc          Erwinia          Escherichia
##          0.000000e+00          3.224379e-03          2.928614e-04
##          Flavobacterium      Gammaproteobacteria_spc          Gilliamella
```

##	1.070853e-05	0.000000e+00	0.000000e+00
##	Lactobacillus	Methylobacterium	Microbacteriaceae_spc
##	0.000000e+00	0.000000e+00	2.868577e-04
##	Microbacterium	Novosphingobium	Ochrobactrum
##	5.906694e-03	3.266800e-05	4.830418e-02
##	Olivibacter	Pedobacter	Phyllobacterium
##	1.615453e-03	1.438613e-03	1.639370e-03
##	Pseudomonas	Pseudoxanthomonas	Rhizobiales_spc
##	5.465078e-05	5.467030e-01	0.000000e+00
##	Roseomonas	Rubrobacter	Salmonella
##	1.093016e-04	0.000000e+00	6.056478e-04
##	Sphingobacteriaceae_spc	Sphingobacteriia_spc	Sphingobacterium
##	9.349097e-04	0.000000e+00	1.008274e-03
##	Staphylococcus	Stenotrophomonas	Veillonella
##	9.007386e-05	1.633400e-05	1.093016e-04
##	Wolbachia	Xanthomonadaceae_spc	Xanthomonadales_spc
##	3.821145e-01	0.000000e+00	0.000000e+00
##	Xanthomonas		
##	0.000000e+00		

```
SD.control <- tapply(control$Abundance, control$Genus, sd)
SD.control
```

##	Acinetobacter	Aeromicrobium	Azospirillum
##	1.369590e-02	2.044845e-04	0.000000e+00
##	Bacilli_spc	Bacillus	Bacteroidetes_spc
##	0.000000e+00	2.341156e-04	2.242407e-03
##	Chryseobacterium	Demetria	Enterobacteriaceae_spc
##	3.343430e-04	3.476237e-03	0.000000e+00
##	Enterobacteriales_spc	Erwinia	Escherichia
##	0.000000e+00	8.859334e-03	1.008105e-03
##	Flavobacterium	Gammaproteobacteria_spc	Gilliamella
##	4.006765e-05	0.000000e+00	0.000000e+00
##	Lactobacillus	Methylobacterium	Microbacteriaceae_spc
##	0.000000e+00	0.000000e+00	7.409020e-04
##	Microbacterium	Novosphingobium	Ochrobactrum
##	1.162316e-02	1.222324e-04	1.750060e-01
##	Olivibacter	Pedobacter	Phyllobacterium
##	4.026769e-03	4.261139e-03	4.698650e-03
##	Pseudomonas	Pseudoxanthomonas	Rhizobiales_spc
##	2.044845e-04	4.674769e-01	0.000000e+00
##	Roseomonas	Rubrobacter	Salmonella
##	4.089690e-04	0.000000e+00	2.031229e-03
##	Sphingobacteriaceae_spc	Sphingobacteriia_spc	Sphingobacterium
##	3.474622e-03	0.000000e+00	3.048420e-03
##	Staphylococcus	Stenotrophomonas	Veillonella
##	3.370255e-04	6.111622e-05	4.089690e-04
##	Wolbachia	Xanthomonadaceae_spc	Xanthomonadales_spc
##	4.686004e-01	0.000000e+00	0.000000e+00
##	Xanthomonas		
##	0.000000e+00		

edit object for plotting

```

#control vs. removal
Bacteria_Genus1$Genus<-as.character(Bacteria_Genus1$Genus)
Bacteria_Genus1$Genus[Bacteria_Genus1$Abundance<0.05]<-"Others"
Bacteria_Genus1$Class<-as.character(Bacteria_Genus1$Class)
Bacteria_Genus1$Class[Bacteria_Genus1$Abundance<0.05]<-"Others"
Bacteria_Genus1$Genus<-factor(Bacteria_Genus1$Genus)
Bacteria_Genus1<-droplevels(Bacteria_Genus1)
Bacteria_Genus1$Treatment<-factor(Bacteria_Genus1$Treatment,levels=c("control", "removal"))
Bacteria_Genus1$Genus <- factor(Bacteria_Genus1$Genus, levels = c("Others", "Ochrobactrum", "Wolbachia"))

#removal vs. 2nd attempt
Bacteria_Genus2$Genus<-as.character(Bacteria_Genus2$Genus)
Bacteria_Genus2$Genus[Bacteria_Genus2$Abundance<0.05]<-"Others"
Bacteria_Genus2$Class<-as.character(Bacteria_Genus2$Class)
Bacteria_Genus2$Class[Bacteria_Genus2$Abundance<0.05]<-"Others"
Bacteria_Genus2$Genus<-factor(Bacteria_Genus2$Genus)
Bacteria_Genus2<-droplevels(Bacteria_Genus2)
Bacteria_Genus2$Treatment<-factor(Bacteria_Genus2$Treatment,levels=c("removal", "2nd-foundation"))
Bacteria_Genus2$Genus <- factor(Bacteria_Genus2$Genus, levels = c("Others", "Ochrobactrum", "Wolbachia"))

```

define color bar

```

Plot_colors_g <- c("grey87", "indianred1", "brown4", "green4", "palegreen", "seagreen")
Plot_colors_g2 <- c("grey87", "indianred1", "brown4", "palegreen", "seagreen")

```

```

Bac <- Bacteria_Genus1
Bac$Sample <- factor(Bac$Sample, levels = c("B0-24", "B0-27", "B0-19a", "B0-22a", "B15-06", "B15-11", "B15-12"))

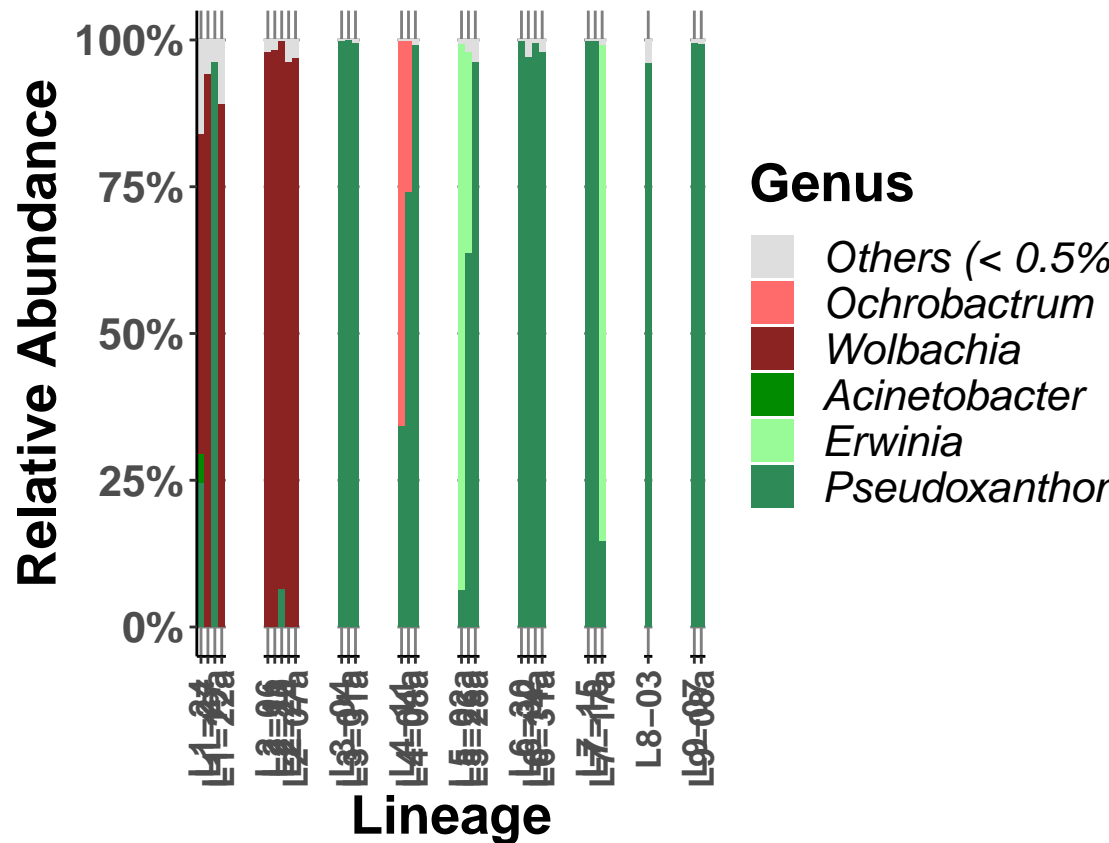
Bacteria_Genus_plot <- ggplot(Bac, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", color = NA, position="fill") +
  scale_fill_manual(values = Plot_colors_g, name = "Genus")

relAB_bac <-Bacteria_Genus_plot +
  facet_grid(~ Linage, scales = "free_x", space = "free_x")+
  #theme(plot.title = element_text(size = 20, face = "bold")) +
  theme_classic()+ #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x="Lineage", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(axis.text.x=element_text(size = 13, angle = 90, vjust = 0.5))+
  theme(text = element_text(size=20, face = "bold"))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 16))+
  theme(legend.text = element_text(face = "italic"))+
  theme(strip.text.x = element_blank())

relAB_bac <- relAB_bac + theme(panel.spacing.x = unit(0.5, "cm"))

relAB_bac

```



plot control vs. removal

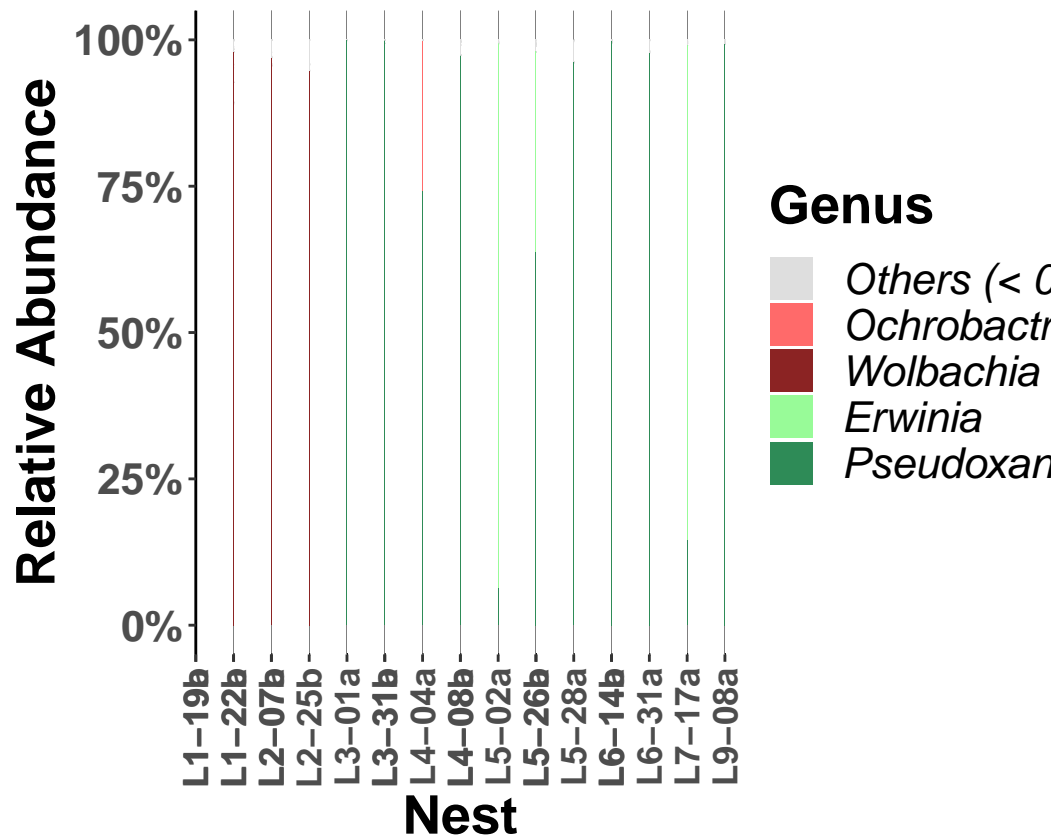
```
Bac2 <- Bacteria_Genus2
Bac2$Sample <- factor(Bac2$Sample, levels = c("B0-19a", "B0-19b", "B0-22a", "B0-22b", "B15-07a", "B15-07b"))

Bacteria_Genus2_plot2 <- ggplot(Bac2, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", color = NA, position="fill") +
  scale_fill_manual(values = Plot_colors_g2, name = "Genus")

g4<-Bacteria_Genus2_plot2 +
  facet_grid(~ Nest, scales = "free_x", space = "free_x")+
  #theme(plot.title = element_text(size = 20, face = "bold")) +
  theme_classic()+
  #gets rid of background
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x="Nest", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(axis.text.x=element_text(size = 13, angle = 90, vjust = 0.5))+
  theme(text = element_text(size=20, face = "bold"))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 16))+
  theme(legend.text = element_text(face = "italic"))+
  theme(strip.text.x = element_blank())

relAb_bac2 <- g4 + theme(panel.spacing.x = unit(0.5, "cm"))

relAb_bac2
```



plot removal vs. 2nd-attempt

Microbiome composition

Composition heatmaps plot control vs. removal (>5% abundance)

carefull, plot was created with fungal data additionally

load fungal data and merge with bacteria

```
CR_fun <- readRDS("//NAS/home/Analysis Removal Experiment/LSU_stand_13.04.22/CR_fun.rds")
rel.CR_fun <- CR_fun %>%
  transform_sample_counts(function(x) {x/sum(x)} )

R2nd_fun <- readRDS("//NAS/home/Analysis Removal Experiment/LSU_stand_13.04.22/R2nd_fun.rds")
rel.R2nd_fun <- R2nd_fun %>%
  transform_sample_counts(function(x) {x/sum(x)} )

#Bacteria
pseqB <- aggregate_rare(rel.CR, level = "Species", detection = 5/100, prevalence = 10/100)
pseqBh <- pseqB %>%
  psmelt()

#Fungi
pseqF <- aggregate_rare(rel.CR_fun, level = "Species", detection = 5/100, prevalence = 10/100)
pseqFh <- pseqF %>%
  psmelt()
```

combine both dataframes and prepare for plotting

```
pseqh <- bind_rows(pseqBh, pseqFh)

pseqh$Treatment<-factor(pseqh$Treatment,levels=c("control", "removal"))
pseqh$Group<-factor(pseqh$Group,levels=c("Bacteria", "Fungi"))
pseqh$Species<-factor(pseqh$Species,levels=c("Wolbachia_spc", "Pseudoxanthomonas_spadix", "Erwinia_spc")
pseqh$Sample <- factor(pseqh$Sample, levels = c("B0-24", "B0-27", "B0-19a", "B0-22a", "B15-06", "B15-11
```

plot heatmap

```
p.heat <- ggplot(pseqh, aes(x = Sample, y = Species)) + geom_tile(aes(fill = Abundance))

# Change color
p.heat <- p.heat + scale_fill_distiller("Abundance", palette = "RdYlBu", labels = scales::percent) + th

# Make bacterial names italics
p.heat <- p.heat + theme(axis.text.y = element_text(colour = 'black',
                                                    size = 10,
                                                    face = 'italic'))

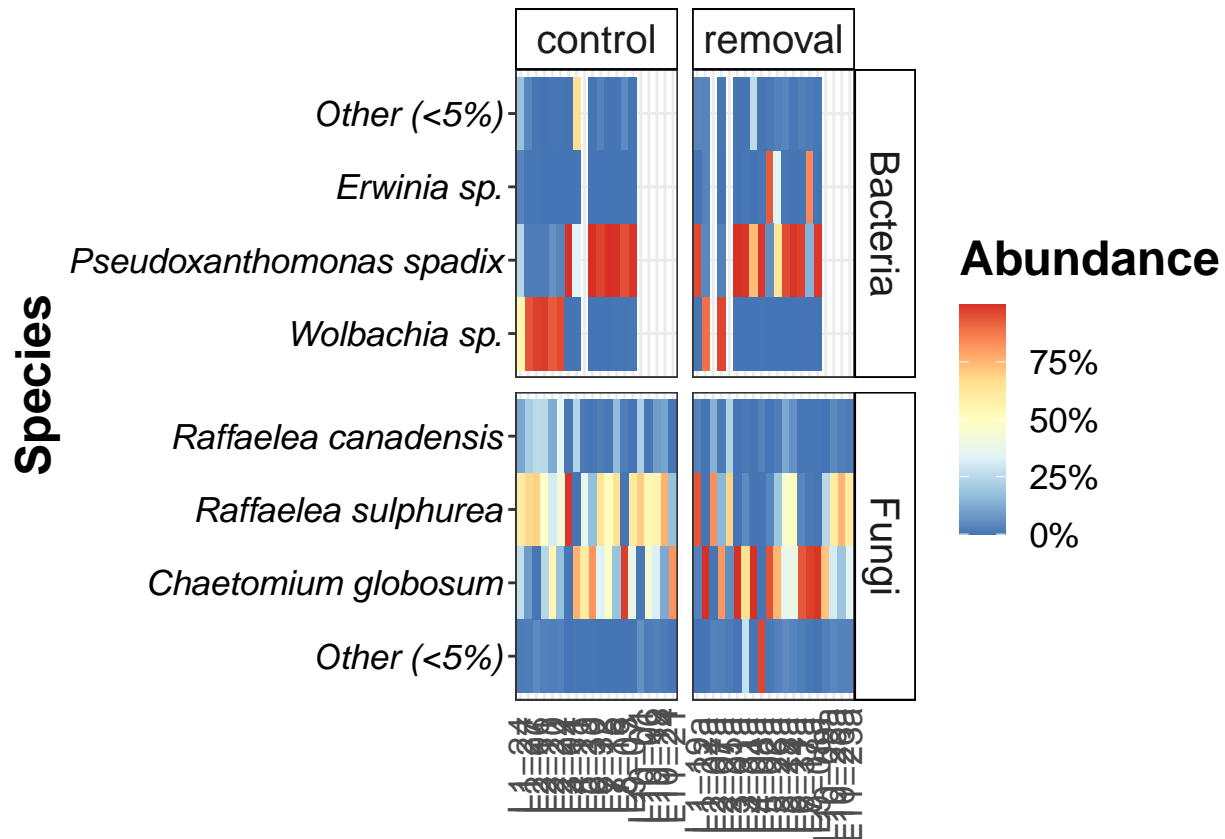
# Make separate samples based on main variable
p.heat <- p.heat + facet_grid(Group ~ Treatment, scales = "free", space = "free") + rremove("x.text")

#Clean the x-axis
p.heat <- p.heat + theme(axis.title.x=element_blank(),
                        axis.text.x=element_blank(),
                        axis.ticks.x=element_blank())

# Clean the facet label box
p.heat <- p.heat + theme(legend.key = element_blank(),
                        strip.background = element_rect(colour="black", fill="white"))+
  theme(strip.text = element_text(size = 15))

p.heat <- p.heat + theme(axis.text.x=element_text(angle = 90, size = 13))+
  theme(axis.title.y = element_text(size = 18, face = "bold"), axis.title.x = element_blank())+
  theme(axis.text.y = element_text(size = 13, face = "italic"))+
  theme(legend.text = element_text(size = 13))+
  theme(legend.title = element_text(size = 18, face = "bold"))

p.heat
```



plot control vs. 2nd attempt (>5% abundance)

```
pseqh2 <- aggregate_rare(rel.R2nd, level = "Species", detection = 5/100, prevalence = 10/100)
pseqh2 <- pseqh2 %>%
  psmelt()

pseqh2 <- aggregate_rare(rel.R2nd_fun, level = "Species", detection = 5/100, prevalence = 10/100)
pseqh2 <- pseqh2 %>%
  psmelt()

pseqh2 <- bind_rows(pseqh2, pseqh2)

pseqh2$Treatment<-factor(pseqh2$Treatment,levels=c("removal", "2nd-foundation"))
pseqh2$Group<-factor(pseqh2$Group,levels=c("Bacteria", "Fungi"))
pseqh2$Species<-factor(pseqh2$Species,levels=c("Wolbachia_spc", "Pseudoxanthomonas_spadix", "Erwinia_"))
pseqh2$Sample <- factor(pseqh2$Sample, levels = c("B0-19a", "B0-19b", "B0-22a", "B0-22b", "B15-01a", "B15-01b"))

p.heat <- ggplot(pseqh2, aes(x = Sample, y = Species)) + geom_tile(aes(fill = Abundance))

# Change color
p.heat <- p.heat + scale_fill_distiller("Abundance", palette = "RdYlBu", labels = scales::percent) + theme_minimal()

# Make bacterial names italics
p.heat <- p.heat + theme(axis.text.y = element_text(colour = 'black',
  size = 10,
  face = 'italic'))
```



```

# Make separate samples based on main variable
p.heat <- p.heat + facet_grid(Group ~ Treatment, scales = "free", space = "free") + rremove("x.text")

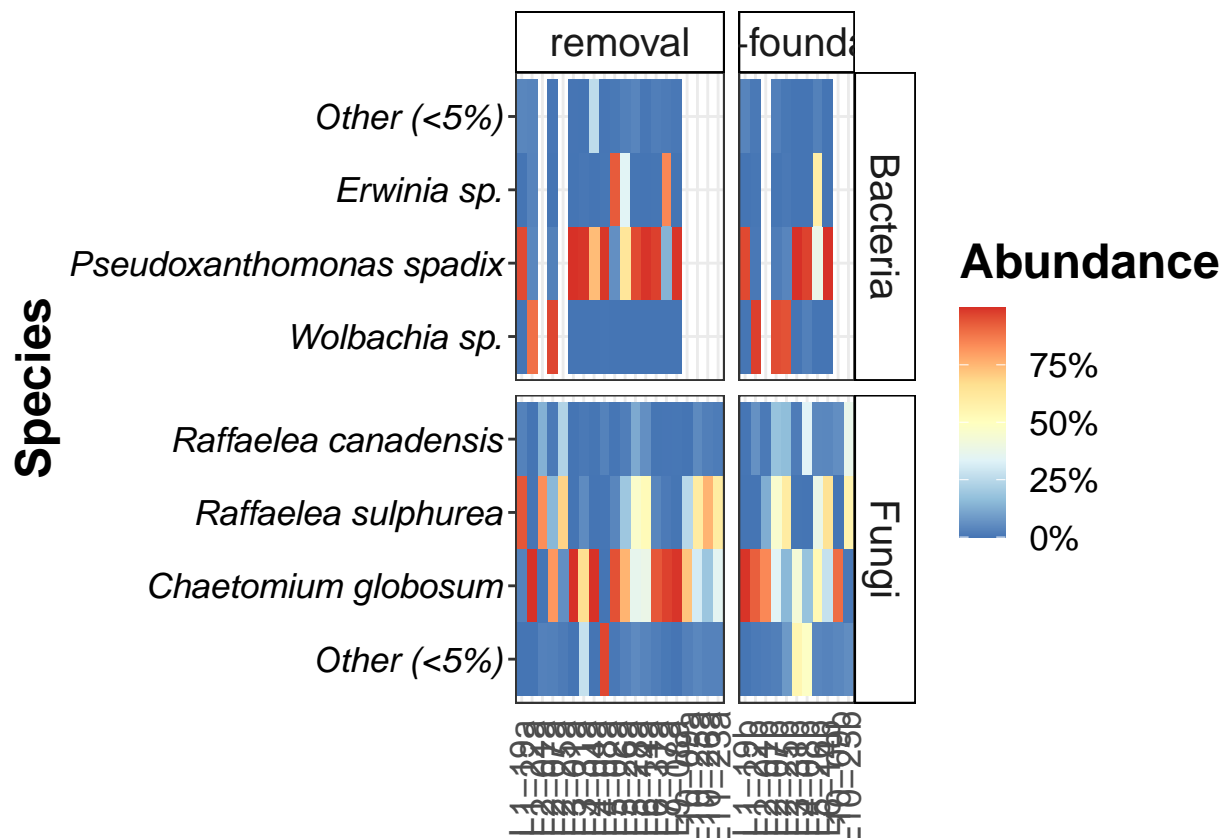
#Clean the x-axis
p.heat <- p.heat + theme(axis.title.x=element_blank(),
                        axis.text.x=element_blank(),
                        axis.ticks.x=element_blank())

# Clean the facet label box
p.heat <- p.heat + theme(legend.key = element_blank(),
                        strip.background = element_rect(colour="black", fill="white"))+
                        theme(strip.text = element_text(size = 15))

p.heat <- p.heat + theme(axis.text.x=element_text(angle = 90, size = 13))+
                        theme(axis.title.y = element_text(size = 18, face = "bold"), axis.title.x = element_blank())+
                        theme(axis.text.y = element_text(size = 13, face = "italic"))+
                        theme(legend.text = element_text(size = 13))+
                        theme(legend.title = element_text(size = 18, face = "bold"))

p.heat

```



Core taxa abundance plot define color bar

```
colpal_bac <- c("brown4", "indianred1", "seagreen", "palegreen")
colpal_bac2 <- c("indianred1", "seagreen", "palegreen")
```

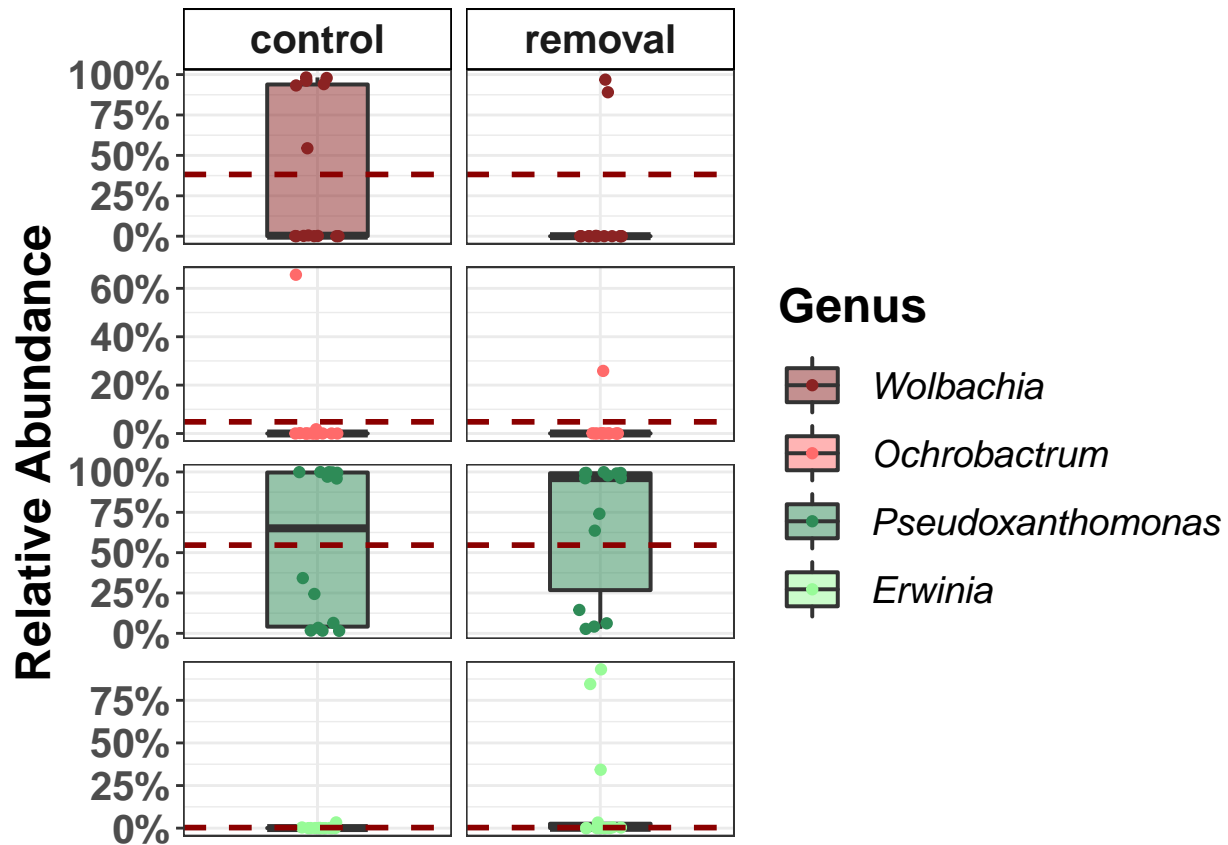
plot core taxa of control vs. removal with relative abundance

```
ps4 <- prune_taxa(taxa_sums(rel.CR) > 0, rel.CR)
ps4 <- tax_glom(ps4, taxrank = 'Genus')
psOrd4 = subset_taxa(ps4, Genus=="Wolbachia" | Genus=="Pseudoxanthomonas" | Genus=="Ochrobactrum" | Genus=="Erwinia")
psctr = subset_samples(psOrd4, Treatment=="control")
#Melt and plot
melt3<-psmelt(psOrd4)
melt4<-psmelt(psctr)
melt3$Treatement <- factor(melt3$Treatment, levels = c("control","removal"))
melt3$Genus <- factor(melt3$Genus, levels = c("Wolbachia", "Ochrobactrum", "Pseudoxanthomonas", "Erwinia"))
melt4$Treatement <- factor(melt4$Treatment, levels = c("control","removal"))
melt4$Genus <- factor(melt4$Genus, levels = c("Wolbachia", "Ochrobactrum", "Pseudoxanthomonas", "Erwinia"))
a_mean <- melt4 %>%
  group_by(Genus) %>%
  summarize(mean_val = mean(Abundance))
print(a_mean)
```

```
## # A tibble: 4 x 2
##   Genus          mean_val
##   <fct>          <dbl>
## 1 Wolbachia      0.382
## 2 Ochrobactrum   0.0483
## 3 Pseudoxanthomonas 0.547
## 4 Erwinia       0.00322
```

```
p2<-ggplot(data = melt3, aes(x = Treatment, y = Abundance)) +
  geom_boxplot(aes(fill=Genus),alpha=0.5,lwd=0.7, position = position_dodge(width = 0.3), width=0.45,ou
  scale_fill_manual(values = colpal_bac, name = "Genus")+
  scale_color_manual(values = colpal_bac, name = "Genus")+
  labs(x = "", y = "Abundance\n")+
  facet_grid(Genus~fct_relevel(Treatment, "control", "removal"), scales = "free")+theme_bw()
p2<-p2+ theme(legend.position="right")+ylab("Relative Abundance")
p2<-p2+ theme(legend.text=element_text(size=14, face = "italic"))+
  theme(legend.key = element_rect(color = NA, fill = NA),legend.key.size = unit(0.9, "cm"))+
  theme(legend.title = element_text(size = 18, face = "bold"))+
  scale_y_continuous(labels=percent_format())
abu2<-p2 + theme(strip.background =element_rect(fill="white", color="black"))+
  theme(strip.text.x = element_text(size = 15, face = "bold"))
Treatment2<-abu2+theme(axis.title.y = element_text(size=18, face="bold"))+theme(axis.text.y = element_t
  theme(axis.text.x = element_text(size=18, angle=90,face="bold"))+
  theme(axis.title.x = element_blank(), axis.text.x=element_blank(),
    axis.ticks.x=element_blank())+
  theme(strip.text.y = element_blank())
relABcoreCR <- Treatment2 + theme(panel.spacing.y = unit(0.3, "cm"))

relABcoreCR
```



plot core taxa of removal vs. 2nd attempt with relative abundance

```
Treats <- c('removal' = "removal", '2nd-foundation' = "2nd-att.")
ps4 <- prune_taxa(taxa_sums(rel.R2nd) > 0, rel.R2nd)
ps4 <- tax_glom(ps4, taxrank = 'Genus')
psOrd4 = subset_taxa(ps4, Genus=="Wolbachia" | Genus=="Pseudoxanthomonas" | Genus=="Ochrobactrum" | Genus=="Erwinia")
psctr = subset_samples(psOrd4, Treatment=="removal")
#Melt and plot
melt<-psmelt(psOrd4)
melt2<-psmelt(psctr)
melt$Treatment <- factor(melt$Treatment, levels = c("removal", "2nd-foundation")) #, labels = c("removal", "2nd-foundation")
melt$Genus <- factor(melt$Genus, levels = c("Wolbachia", "Ochrobactrum", "Pseudoxanthomonas", "Erwinia"))
melt2$Treatment <- factor(melt2$Treatment, levels = c("removal", "2nd-foundation")) #, labels = c("removal", "2nd-foundation")
melt2$Genus <- factor(melt2$Genus, levels = c("Wolbachia", "Ochrobactrum", "Pseudoxanthomonas", "Erwinia"))
a_mean <- melt2 %>%
  group_by(Genus) %>%
  summarize(mean_val = mean(Abundance))
print(a_mean)
```

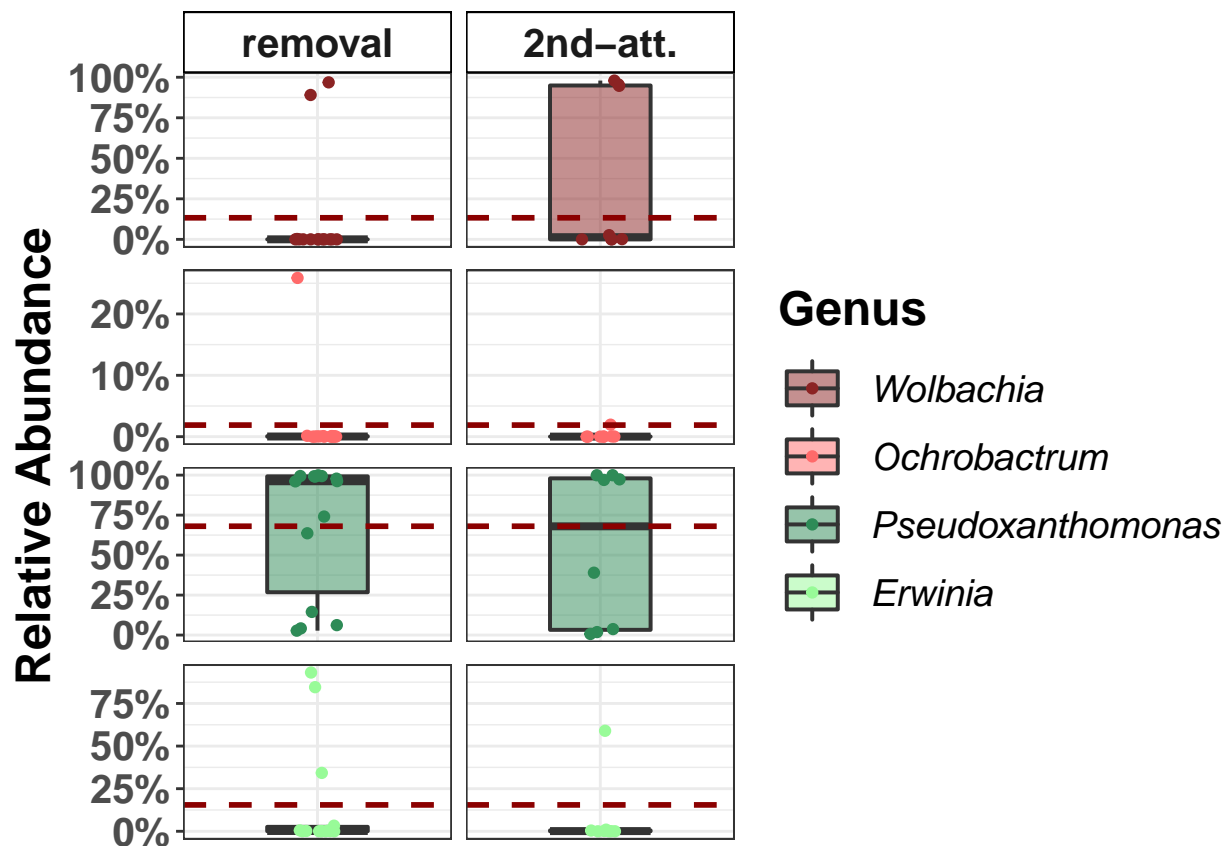
```
## # A tibble: 4 x 2
##   Genus          mean_val
##   <fct>          <dbl>
## 1 Wolbachia      0.133
## 2 Ochrobactrum   0.0189
## 3 Pseudoxanthomonas 0.680
## 4 Erwinia       0.155
```

```

p<-ggplot(data = melt, aes(x = Treatment, y = Abundance)) +
  geom_boxplot(aes(fill=Genus),alpha=0.5,lwd=0.7, position = position_dodge(width = 0.3), width=0.45,ou
  scale_fill_manual(values = colpal_bac, name = "Genus")+
  scale_color_manual(values = colpal_bac, name = "Genus")+
  labs(x = "", y = "Abundance\n")+
  facet_grid(Genus~fct_relevel(Treatment, "removal", "2nd-foundation"), labeller = as_labeller(Treats),
p<-p+ theme(legend.position="right")+ylab("Relative Abundance")
p<-p+ theme(legend.text=element_text(size=14, face = "italic"))+theme(legend.key = element_rect(color =
  scale_y_continuous(labels=percent_format()))
abu<-p + theme(strip.background =element_rect(fill="white", color="black"))+
  theme(strip.text.x = element_text(size = 15, face = "bold"))
TreatmentR2nd<-abu+theme(axis.title.y = element_text(size=18, face="bold"))+theme(axis.text.y = element
  theme(axis.text.x = element_text(size=18, angle=90,face="bold"))+
  theme(axis.title.x = element_blank(), axis.text.x=element_blank(),
    axis.ticks.x=element_blank())+
  theme(strip.text.y = element_blank())
relAbcoreR2nd_bac <- TreatmentR2nd + theme(panel.spacing.y = unit(0.3, "cm"))

relAbcoreR2nd_bac

```



Ordination analysis

Sample ordination

```
set.seed(1)
ordi.CR = ordinate(rel.CR, "NMDS", "bray", k=3, trymax=100)
```

NMDS

```
## Run 0 stress 0.004101445
## Run 1 stress 0.004098865
## ... New best solution
## ... Procrustes: rmse 0.002824601  max resid 0.008262479
## ... Similar to previous best
## Run 2 stress 0.004117503
## ... Procrustes: rmse 0.001517589  max resid 0.003808871
## ... Similar to previous best
## Run 3 stress 0.004076859
## ... New best solution
## ... Procrustes: rmse 0.002901198  max resid 0.01008774
## Run 4 stress 0.004059221
## ... New best solution
## ... Procrustes: rmse 0.002873452  max resid 0.007736466
## ... Similar to previous best
## Run 5 stress 0.004082078
## ... Procrustes: rmse 0.001587272  max resid 0.004786337
## ... Similar to previous best
## Run 6 stress 0.004063495
## ... Procrustes: rmse 0.0007269976  max resid 0.002142316
## ... Similar to previous best
## Run 7 stress 0.006355146
## Run 8 stress 0.004066319
## ... Procrustes: rmse 0.001770016  max resid 0.004658518
## ... Similar to previous best
## Run 9 stress 0.004082557
## ... Procrustes: rmse 0.001916299  max resid 0.005164159
## ... Similar to previous best
## Run 10 stress 0.004104849
## ... Procrustes: rmse 0.001891373  max resid 0.004529891
## ... Similar to previous best
## Run 11 stress 0.00551348
## Run 12 stress 0.004050816
## ... New best solution
## ... Procrustes: rmse 0.002384241  max resid 0.007869152
## ... Similar to previous best
## Run 13 stress 0.004105757
## ... Procrustes: rmse 0.003220953  max resid 0.01171721
## Run 14 stress 0.00411733
## ... Procrustes: rmse 0.004184579  max resid 0.01101352
## Run 15 stress 0.004205084
## ... Procrustes: rmse 0.005207008  max resid 0.01382579
## Run 16 stress 0.004059522
## ... Procrustes: rmse 0.002206135  max resid 0.007141013
## ... Similar to previous best
## Run 17 stress 0.004066419
## ... Procrustes: rmse 0.002952615  max resid 0.009668699
```

```
## ... Similar to previous best
## Run 18 stress 0.004237588
## ... Procrustes: rmse 0.004326122  max resid 0.01153981
## Run 19 stress 0.007574037
## Run 20 stress 0.004062869
## ... Procrustes: rmse 0.002709043  max resid 0.00885975
## ... Similar to previous best
## *** Solution reached
```

```
ordi.CR$stress
```

```
## [1] 0.004050816
```

```
set.seed(1)
ordi.R2nd = ordinate(rel.R2nd, "NMDS", "bray", k=3, trymax=100)
```

```
## Run 0 stress 0.000497131
## Run 1 stress 0.00197463
## Run 2 stress 0.001703256
## Run 3 stress 0.001308364
## Run 4 stress 0.001248164
## Run 5 stress 0.001519522
## Run 6 stress 0.001343547
## Run 7 stress 0.001598612
## Run 8 stress 0.001721315
## Run 9 stress 0.001897331
## Run 10 stress 0.001720918
## Run 11 stress 0.001865885
## Run 12 stress 0.00113327
## Run 13 stress 0.002059709
## Run 14 stress 0.001943087
## Run 15 stress 0.001545742
## Run 16 stress 0.001559676
## Run 17 stress 0.001704741
## Run 18 stress 0.002029385
## Run 19 stress 0.001869968
## Run 20 stress 0.001005818
## Run 21 stress 0.002108947
## Run 22 stress 0.001967605
## Run 23 stress 0.001303404
## Run 24 stress 0.001647096
## Run 25 stress 0.001852263
## Run 26 stress 0.002365609
## Run 27 stress 0.001375455
## Run 28 stress 0.001762234
## Run 29 stress 0.001435289
## Run 30 stress 0.002406851
## Run 31 stress 0.001297131
## Run 32 stress 0.002199374
## Run 33 stress 0.001483006
## Run 34 stress 0.001361984
## Run 35 stress 0.00181785
## Run 36 stress 0.001490938
```

```
## Run 37 stress 0.002538429
## Run 38 stress 0.002129452
## Run 39 stress 0.002098761
## Run 40 stress 0.001472975
## Run 41 stress 0.001515251
## Run 42 stress 0.001708189
## Run 43 stress 0.002163114
## Run 44 stress 0.001851951
## Run 45 stress 0.0008179611
## ... Procrustes: rmse 0.006359988  max resid 0.01984187
## Run 46 stress 0.001648781
## Run 47 stress 0.001419709
## Run 48 stress 0.002341615
## Run 49 stress 0.002229777
## Run 50 stress 0.001347667
## Run 51 stress 0.002041472
## Run 52 stress 0.001705717
## Run 53 stress 0.001793895
## Run 54 stress 0.001430107
## Run 55 stress 0.001735565
## Run 56 stress 0.001664429
## Run 57 stress 0.002675406
## Run 58 stress 0.001438074
## Run 59 stress 0.002363232
## Run 60 stress 0.001713227
## Run 61 stress 0.001733833
## Run 62 stress 0.001718886
## Run 63 stress 0.001257307
## Run 64 stress 0.001601473
## Run 65 stress 0.001702062
## Run 66 stress 0.001763214
## Run 67 stress 0.001740427
## Run 68 stress 0.00202921
## Run 69 stress 0.001591445
## Run 70 stress 0.001584794
## Run 71 stress 0.00144915
## Run 72 stress 0.002198814
## Run 73 stress 0.001753384
## Run 74 stress 0.001338002
## Run 75 stress 0.002008779
## Run 76 stress 0.001476132
## Run 77 stress 0.002114631
## Run 78 stress 0.001935933
## Run 79 stress 0.001514252
## Run 80 stress 0.001411258
## Run 81 stress 0.001563949
## Run 82 stress 0.001646717
## Run 83 stress 0.001543691
## Run 84 stress 0.001656827
## Run 85 stress 0.00196587
## Run 86 stress 0.00180755
## Run 87 stress 0.001456862
## Run 88 stress 0.001799494
## Run 89 stress 0.001490406
```

```
## Run 90 stress 0.001900829
## Run 91 stress 0.001626272
## Run 92 stress 0.00215964
## Run 93 stress 0.001492013
## Run 94 stress 0.00141128
## Run 95 stress 0.002279657
## Run 96 stress 0.002300291
## Run 97 stress 0.00135705
## Run 98 stress 0.002098906
## Run 99 stress 0.001471998
## Run 100 stress 0.0009480887
## ... Procrustes: rmse 0.008738421 max resid 0.02757519
## *** No convergence -- monoMDS stopping criteria:
## 100: no. of iterations >= maxit
```

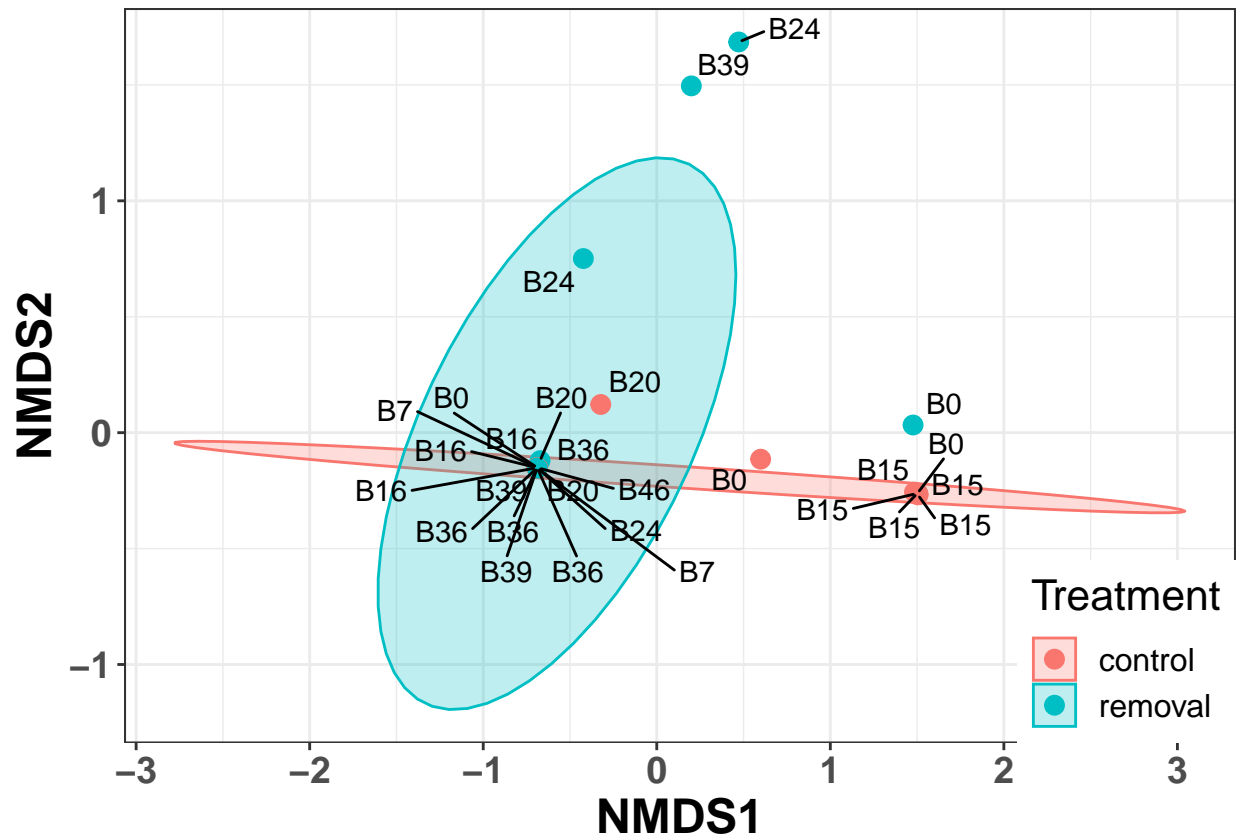
```
ordi.R2nd$stress
```

```
## [1] 0.000497131
```

plot NMDS

```
NMDS_CR_bac <- plot_ordination(rel.CR, ordi.CR, color = "Treatment", axes=c(1,2))+
  stat_ellipse(level = 0.95, geom = "polygon", aes(fill = Treatment), alpha = 0.25)+
  geom_point(size=3, inherit.aes=T)+
  geom_text_repel(aes(label=Linage), color = "black", size = 4, max.overlaps = Inf)+
  theme_bw()+
  theme(legend.justification=c(1,0), legend.position=c(1,0))+
  theme(axis.text = element_text(size = 14, face = "bold"))+
  theme(axis.title = element_text(size = 18, face = "bold"))+
  theme(legend.text = element_text(size = 12))+
  theme(legend.title = element_text(size = 16))+
  theme(title = element_text(size = 18))
```

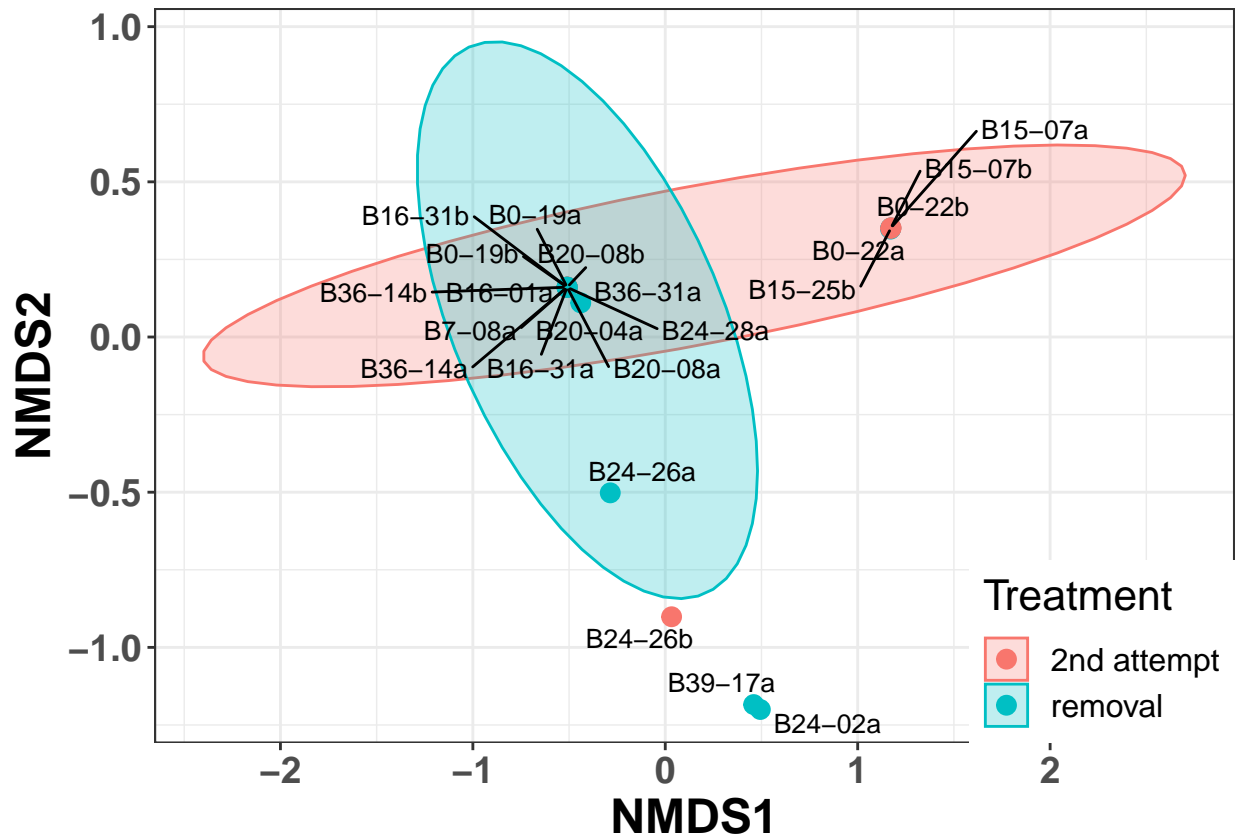
```
NMDS_CR_bac
```

```
sample_data(rel.R2nd)$Treatment <- as.factor(sample_data(rel.R2nd)$Treatment)
levels(sample_data(rel.R2nd)$Treatment) = c("2nd attempt", "removal")

NMDS_R2nd_bac <- plot_ordination(rel.R2nd, ordi.R2nd, color = "Treatment", axes=c(1,2))+
  stat_ellipse(level = 0.95, geom = "polygon", aes(fill = Treatment), alpha = 0.25)+
  geom_point(size=3, inherit.aes=T)+
  geom_text_repel(aes(label=Sample), color = "black", size = 3.5, max.overlaps = Inf)+
  theme_bw()+
  theme(legend.justification=c(1,0), legend.position=c(1,0))+
  theme(axis.text = element_text(size = 14, face = "bold"))+
  theme(axis.title = element_text(size = 18, face = "bold"))+
  theme(legend.text = element_text(size = 12))+
  theme(legend.title = element_text(size = 16))+
  theme(title = element_text(size = 18))

NMDS_R2nd_bac
```



Permanova for community-level multivariate comparisons

PERMANOVA quantifies multivariate community-level differences between groups.

```
# Pick relative abundances (compositional) and sample metadata
otu.CR <- abundances(rel.CR)
meta.CR <- meta(rel.CR)

otu.R2nd <- abundances(rel.R2nd)
meta.R2nd <- meta(rel.R2nd)
```

PERMANOVA significance test for group-level differences Now let us evaluate whether the Treatment has a significant effect on overall garden microbiota composition. Perform PERMANOVA:

```
# control vs. removal
set.seed(1)
adonis2(distance(rel.CR, method = "bray") ~ Treatment + Linage, data = meta.CR)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = distance(rel.CR, method = "bray") ~ Treatment + Linage, data = meta.CR)
```

```
##           Df SumOfSqs      R2      F Pr(>F)
## Treatment  1   0.4260 0.06554 4.2998 0.038 *
## Linage     8   4.2905 0.66007 5.4126 0.001 ***
## Residual   18   1.7835 0.27439
## Total      27   6.5001 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#removal vs. 2nd attempt
```

```
set.seed(1)
adonis2(distance(rel.R2nd, method = "bray") ~ Treatment + Linage, data = meta.R2nd)
```

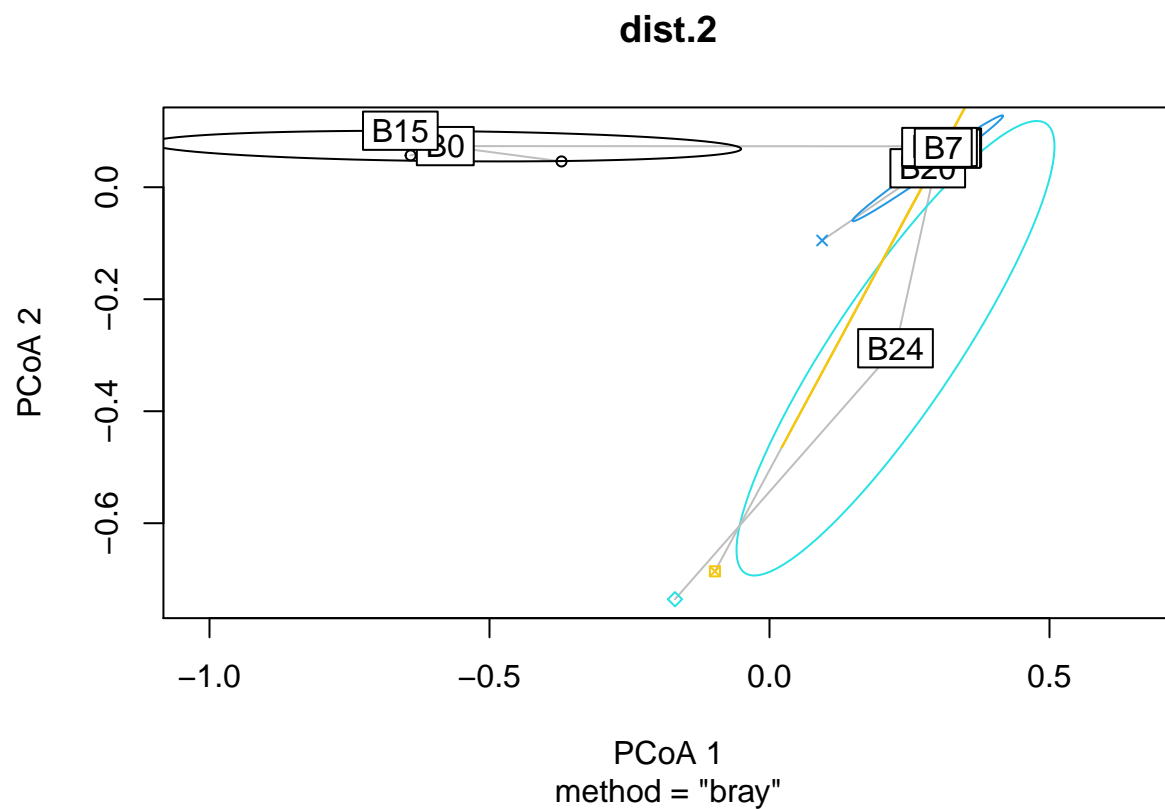
```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = distance(rel.R2nd, method = "bray") ~ Treatment + Linage, data = meta.R2nd)
##           Df SumOfSqs      R2      F Pr(>F)
## Treatment  1   0.2100 0.04241 1.8759 0.169
## Linage     7   3.2869 0.66369 4.1939 0.003 **
## Residual   13   1.4555 0.29390
## Total      21   4.9525 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Checking the homogeneity condition Check that variance homogeneity assumptions hold (to ensure the reliability of the results): Note the assumption of similar multivariate spread among the groups ie. analogous to variance homogeneity. Here the groups have signif. different spreads and permanova result may be potentially explained by that.

```
# control vs. removal
dist <- vegdist(t(otu.CR))
dist.2 <- betadisper(dist, meta.CR$Linage)
anova(dist.2)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      8 0.60721 0.075901  1.1552 0.374
## Residuals   19 1.24843 0.065707
```

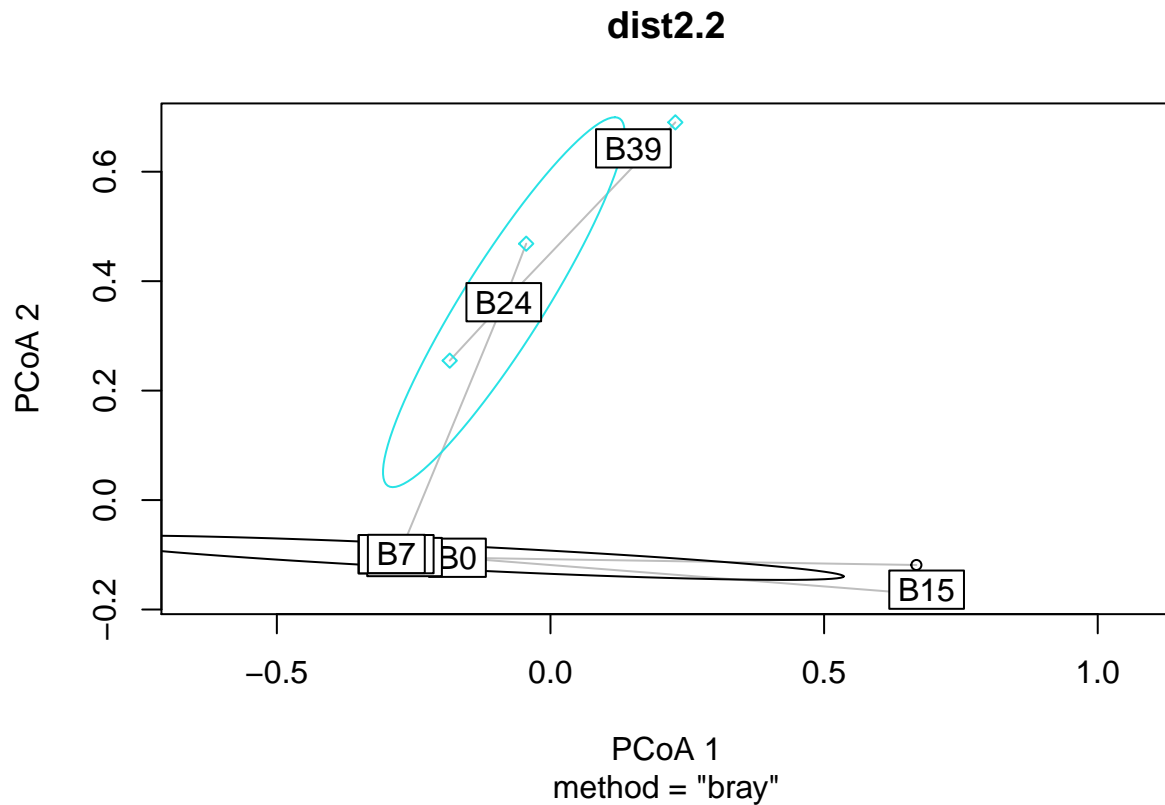
```
plot(dist.2, hull = FALSE, ellipse = TRUE)
```



```
# removal vs. 2nd attempt
dist2 <- vegdist(t(otu.R2nd))
dist2.2 <- betadisper(dist2, meta.R2nd$Linage)
anova(dist2.2)
```

```
## Analysis of Variance Table
##
## Response: Distances
##      Df Sum Sq Mean Sq F value Pr(>F)
## Groups    7 0.76889 0.109841  2.1272 0.1087
## Residuals 14 0.72292 0.051637
```

```
plot(dist2.2, hull = FALSE, ellipse = TRUE)
```



closer look at most abundant taxa

subset dataset Bacteria_Genus of both combinations to most abundant Geni (each *Wolbachia*, *Pseudoxanthomonas*) (high.reads with rel. abundance of total dataset)

```
WolbachiaCR <- subset_taxa(rel.CR, Genus == "Wolbachia")
dfWCR <- WolbachiaCR %>%
  tax_glom(taxrank = "Genus") %>%
  psmelt()

PseudoCR <- subset_taxa(rel.CR, Genus == "Pseudoxanthomonas")
dfPCR <- PseudoCR %>%
  tax_glom(taxrank = "Genus") %>%
  psmelt()

WolbachiaR2nd <- subset_taxa(rel.R2nd, Genus == "Wolbachia")
dfWR2nd <- WolbachiaR2nd %>%
  tax_glom(taxrank = "Genus") %>%
  psmelt()

PseudoR2nd <- subset_taxa(rel.R2nd, Genus == "Pseudoxanthomonas")
dfPR2nd <- PseudoR2nd %>%
```

```
tax_glom(taxrank = "Genus") %>%
psmelt()
```

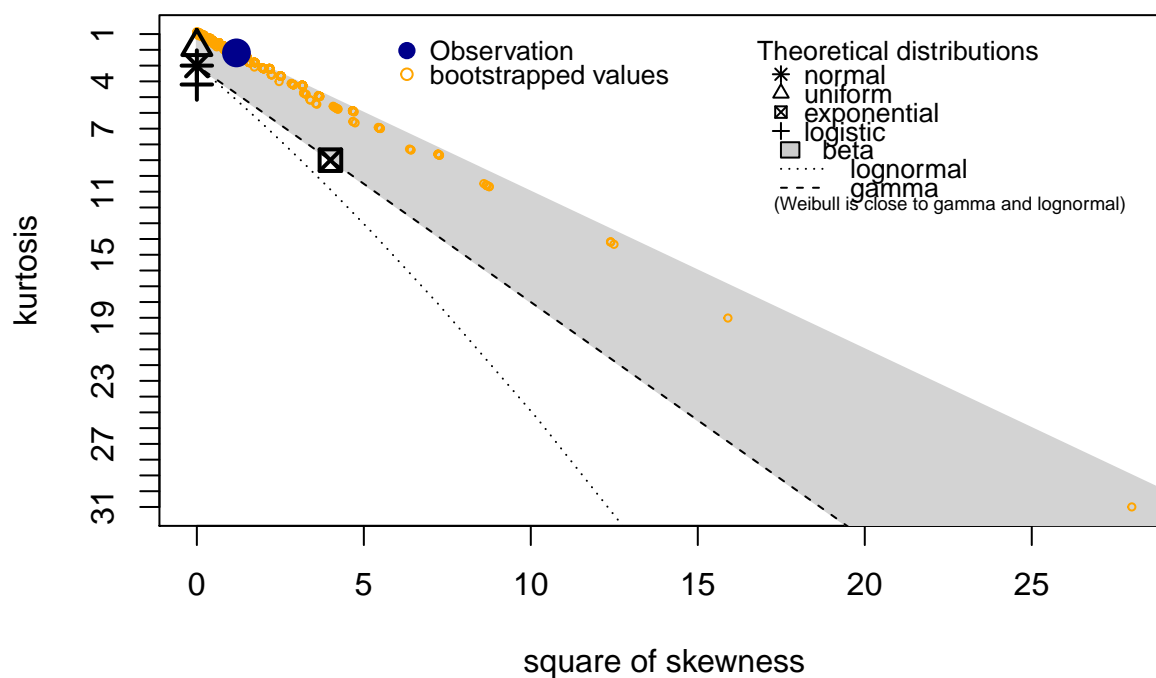
dataset *Wolbachia*

abundance between treatments

Tukey transformation of data

```
descdist(dfWCR$Abundance, boot = 1000)
```

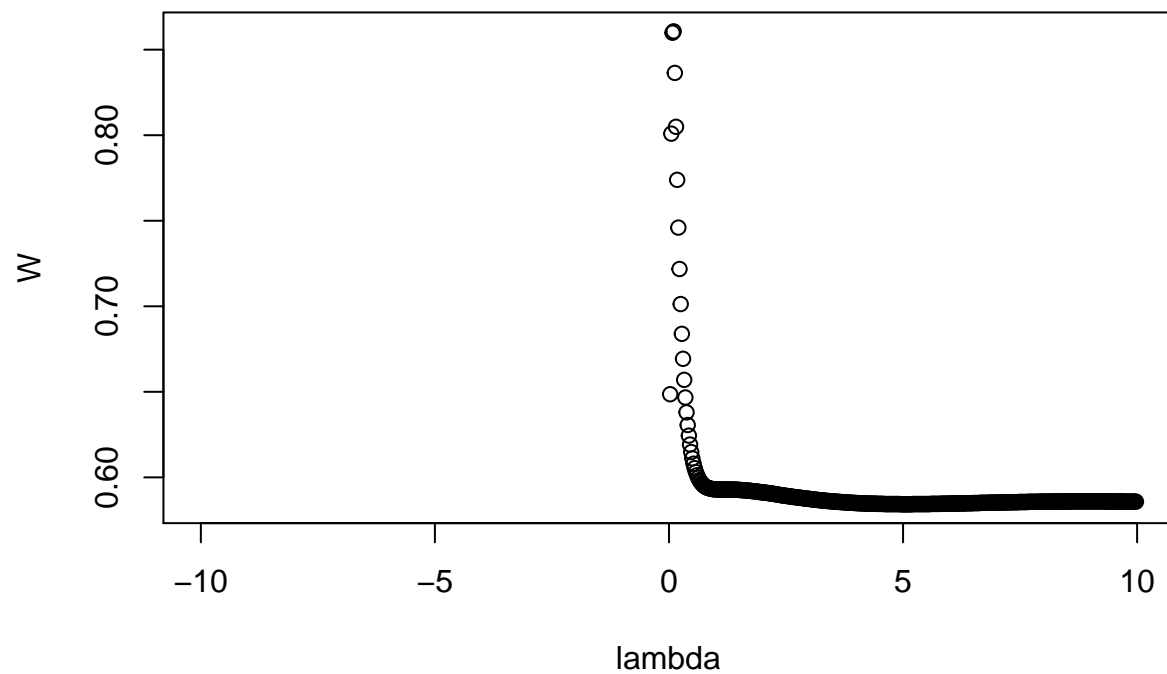
Cullen and Frey graph



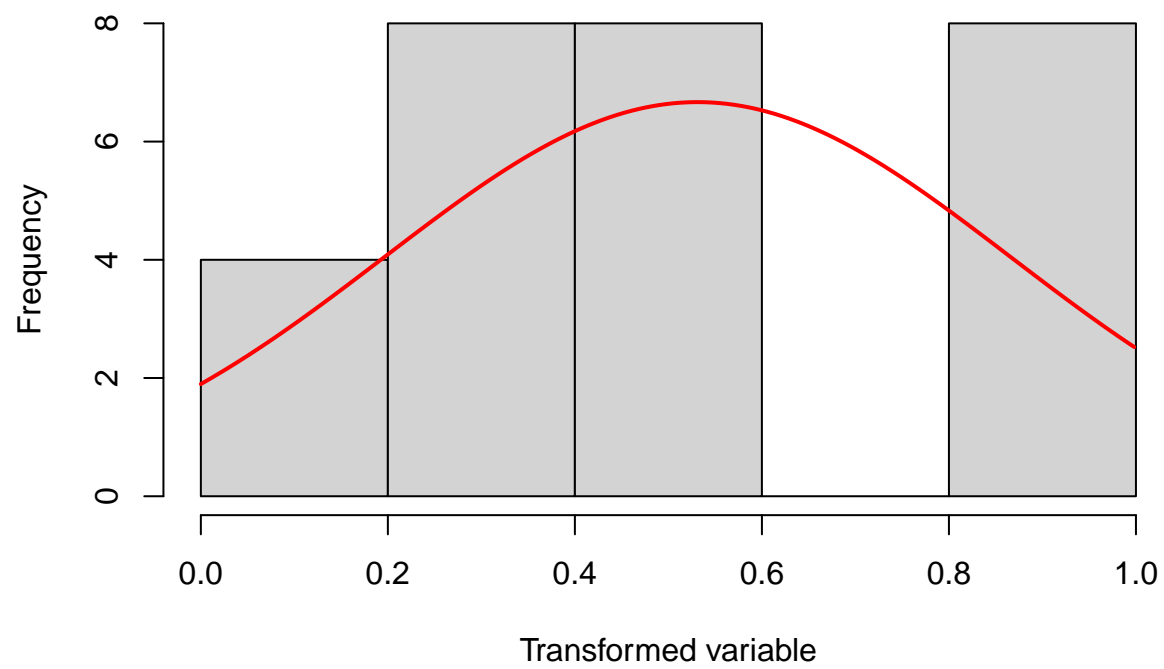
```
## summary statistics
## -----
## min: 0    max: 0.9822504
## median: 0.0001955422
## mean: 0.2575581
## estimated sd: 0.4204394
## estimated skewness: 1.088921
## estimated kurtosis: 2.194203
```

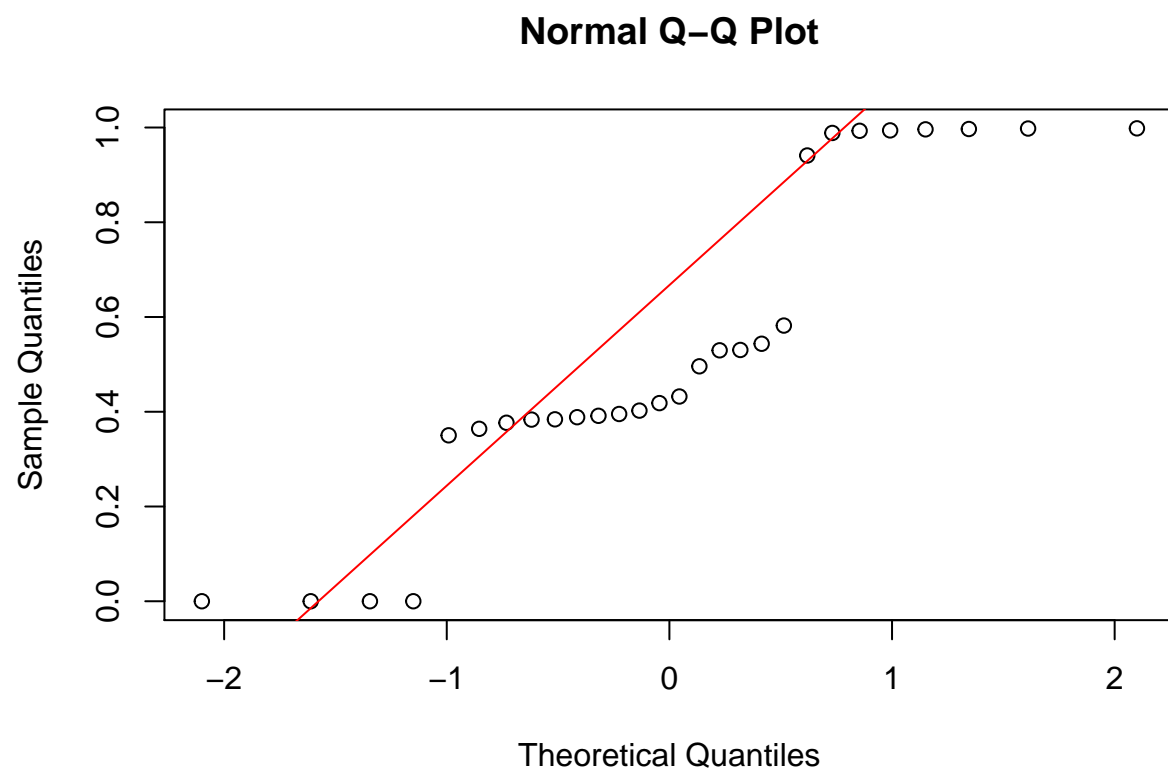
```
library(rcompanion)

dfWCR$Ab.tuk <- rcompanion::transformTukey(dfWCR$Abundance)
```

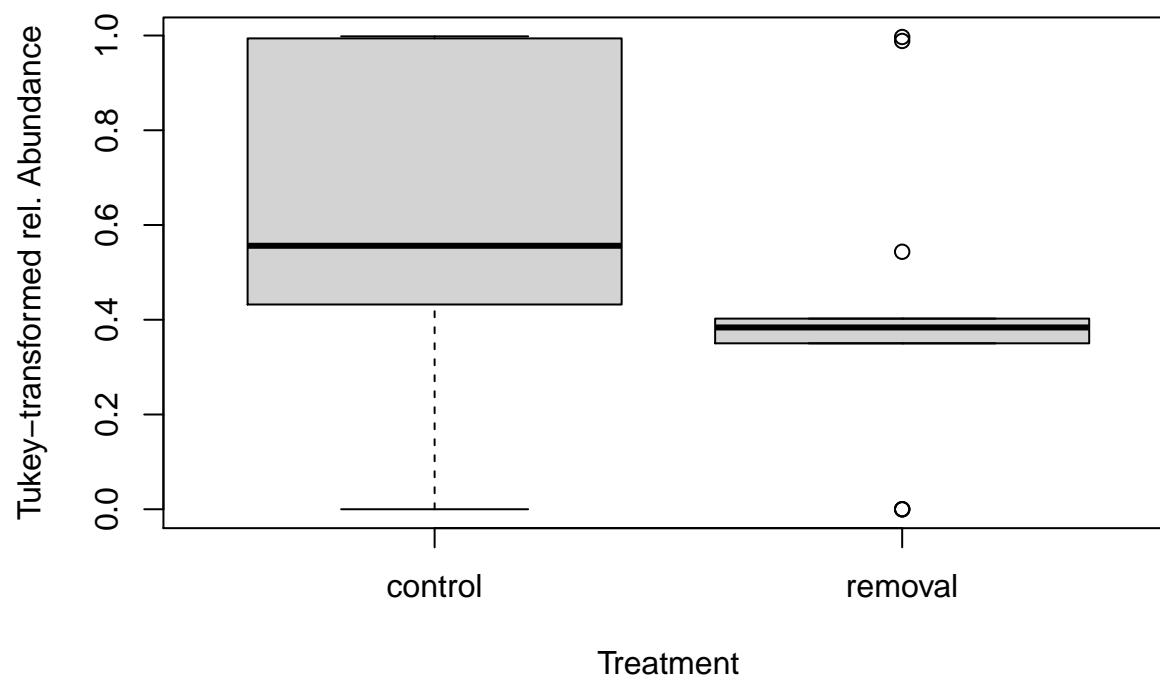


```
##
##      lambda      W Shapiro.p.value
## 405    0.1 0.8607      0.001548
##
## if (lambda > 0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda < 0){TRANS = -1 * x ^ lambda}
```





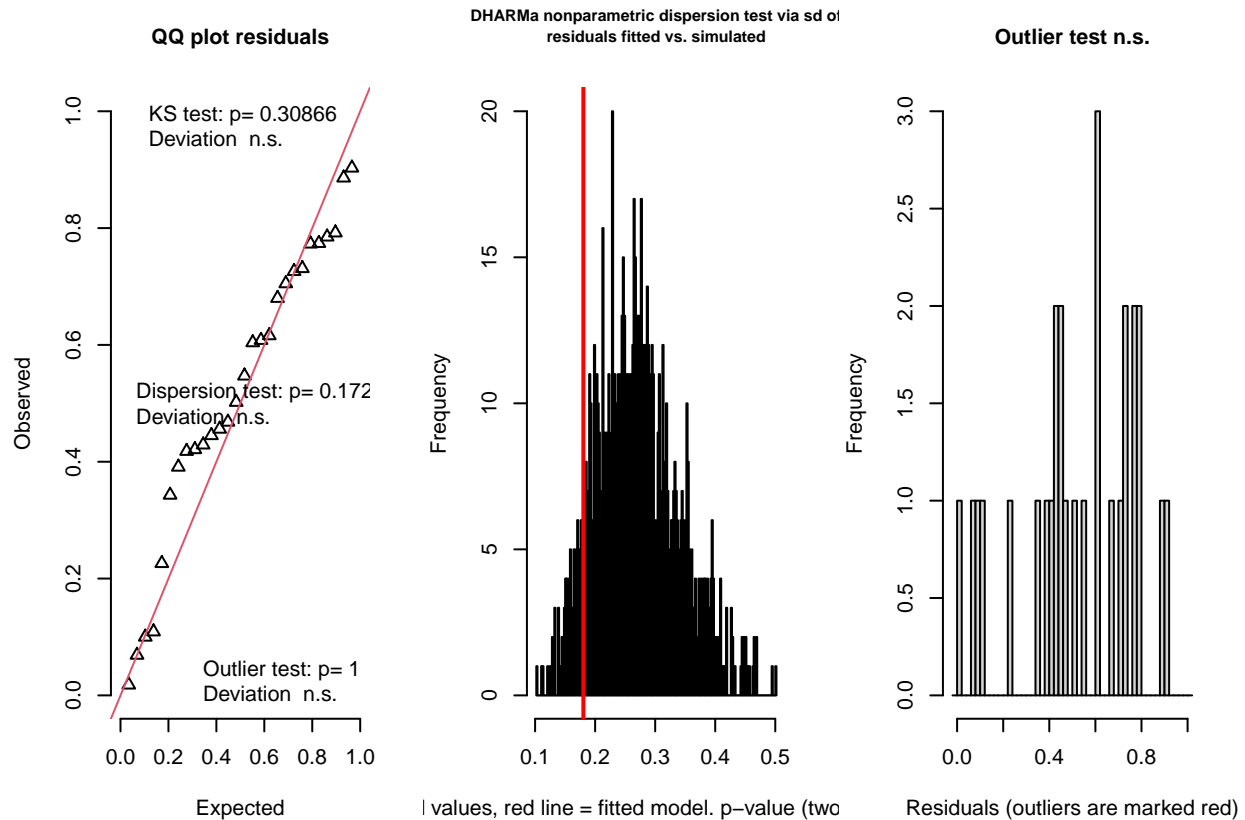
```
boxplot(Ab.tuk ~ Treatment, data = dfWCR, ylab = "Tukey-transformed rel. Abundance", xlab = "Treatment")
```



```
trans <- lm(Ab.tuk ~ Treatment + Linage, data = dfWCR)
Anova(trans, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: Ab.tuk
##          Sum Sq Df F value    Pr(>F)
## Treatment 0.07815  1  2.4061 0.1382692
## Linage     1.95094  8  7.5085 0.0002021 ***
## Residuals 0.58462 18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_trans <- simulateResiduals(trans, n = 1000)
testResiduals(res_trans)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17671, p-value = 0.3087
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.66538, p-value = 0.172
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 28, p-value = 1
```

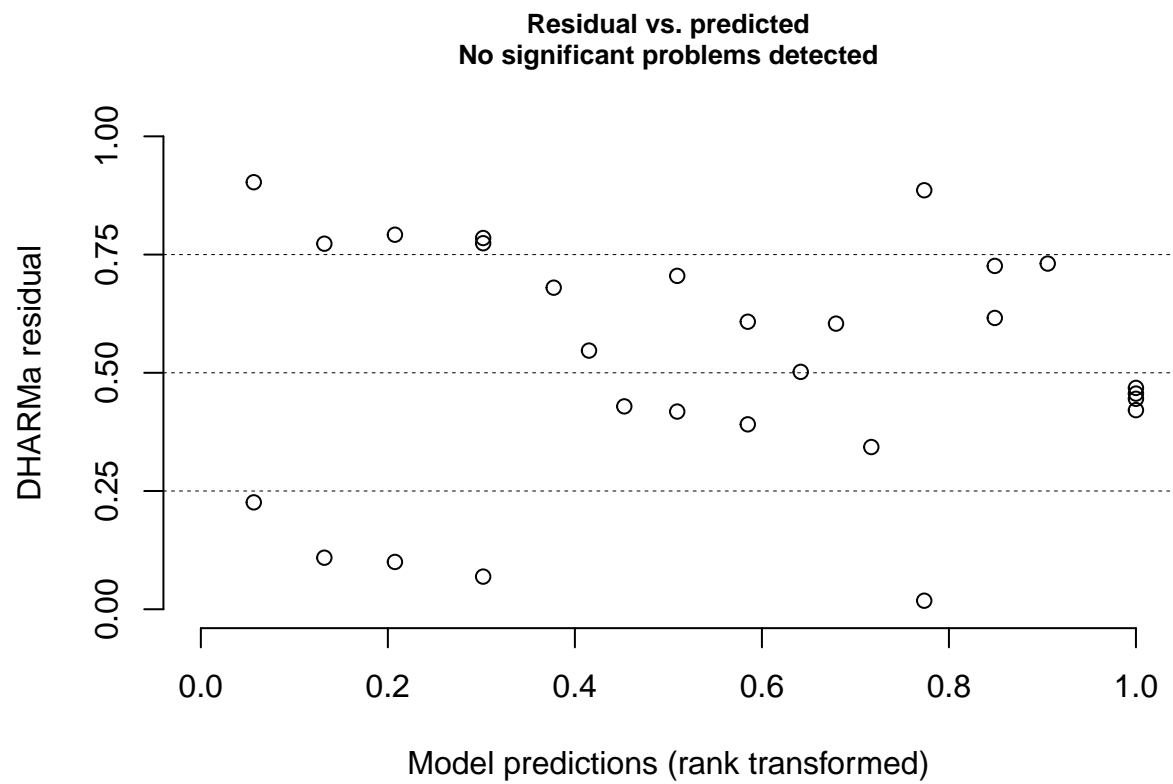
```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1234361
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.17671, p-value = 0.3087
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.66538, p-value = 0.172
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 28, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1234361
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

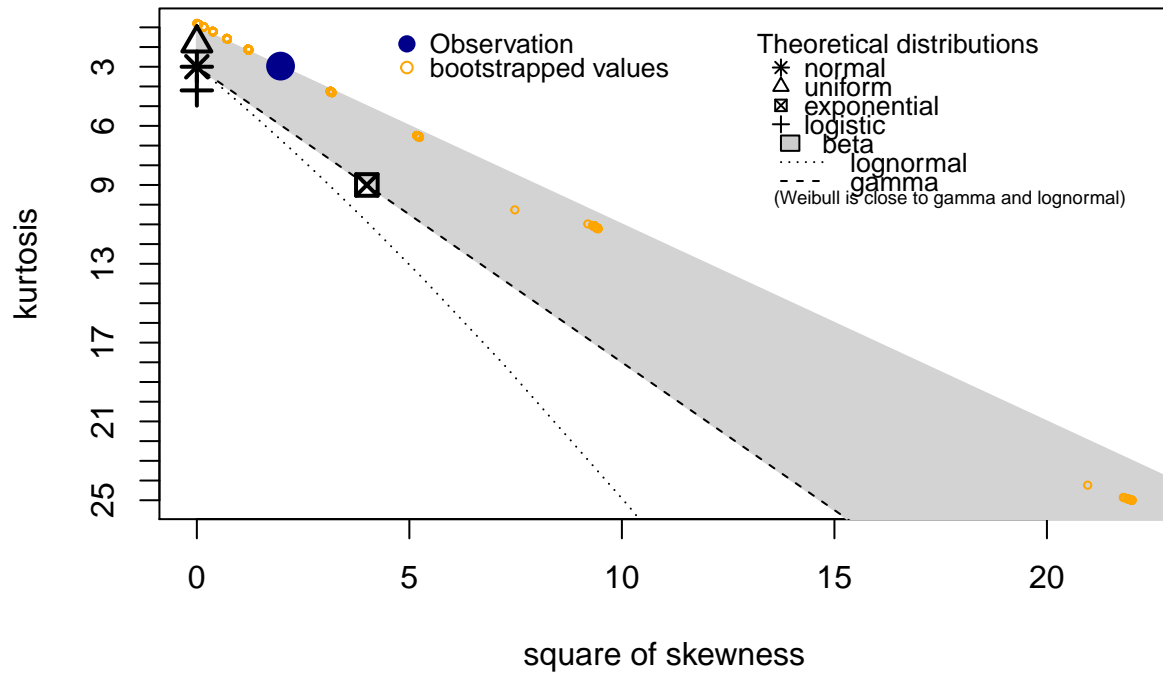
```

```
plotResiduals(res_trans)
```



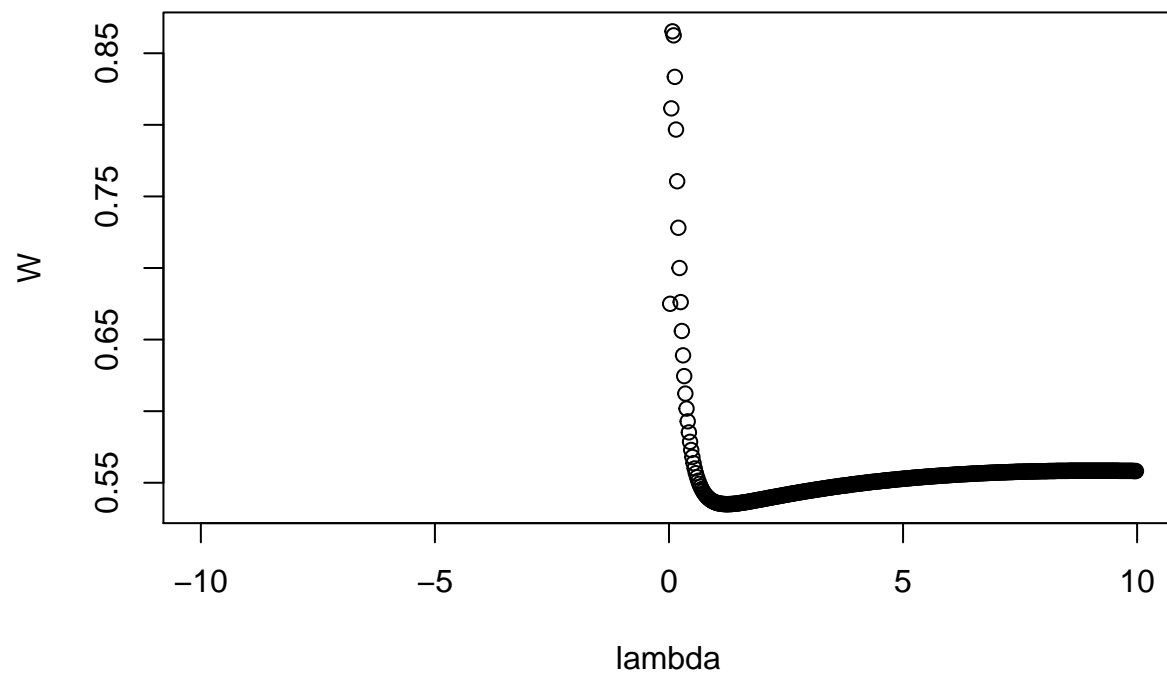
```
descdist(dfWR2nd$Abundance, boot = 1000)
```

Cullen and Frey graph

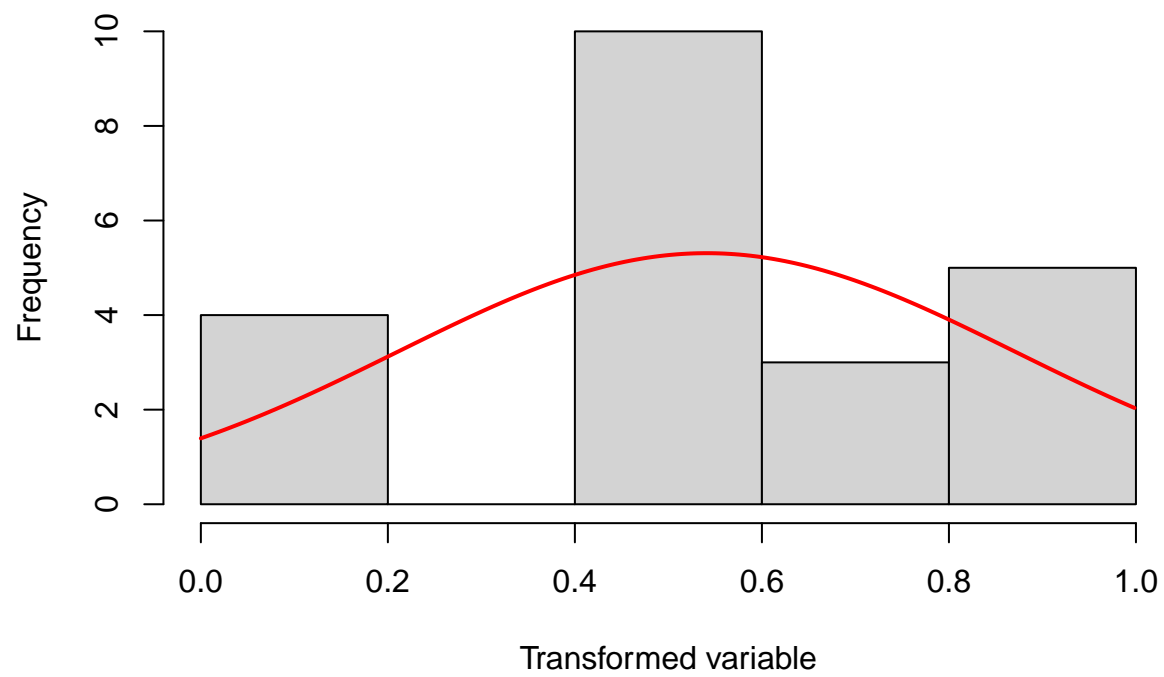


```
## summary statistics
## -----
## min: 0    max: 0.9793903
## median: 9.297361e-05
## mean: 0.2168947
## estimated sd: 0.406304
## estimated skewness: 1.403302
## estimated kurtosis: 2.967066
```

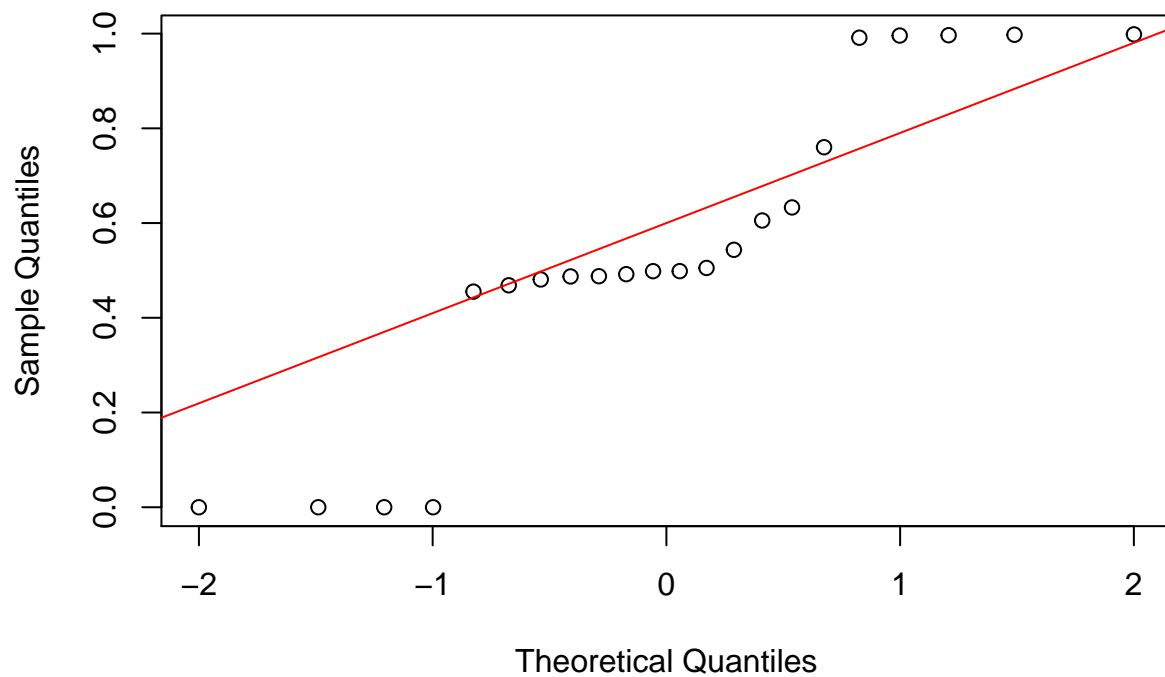
```
dfWR2nd$Ab.tuk <- rcompanion::transformTukey(dfWR2nd$Abundance)
```



```
##
##      lambda      W Shapiro.p.value
## 404  0.075 0.8653          0.006398
##
## if (lambda > 0){TRANS = x ^ lambda}
## if (lambda == 0){TRANS = log(x)}
## if (lambda < 0){TRANS = -1 * x ^ lambda}
```



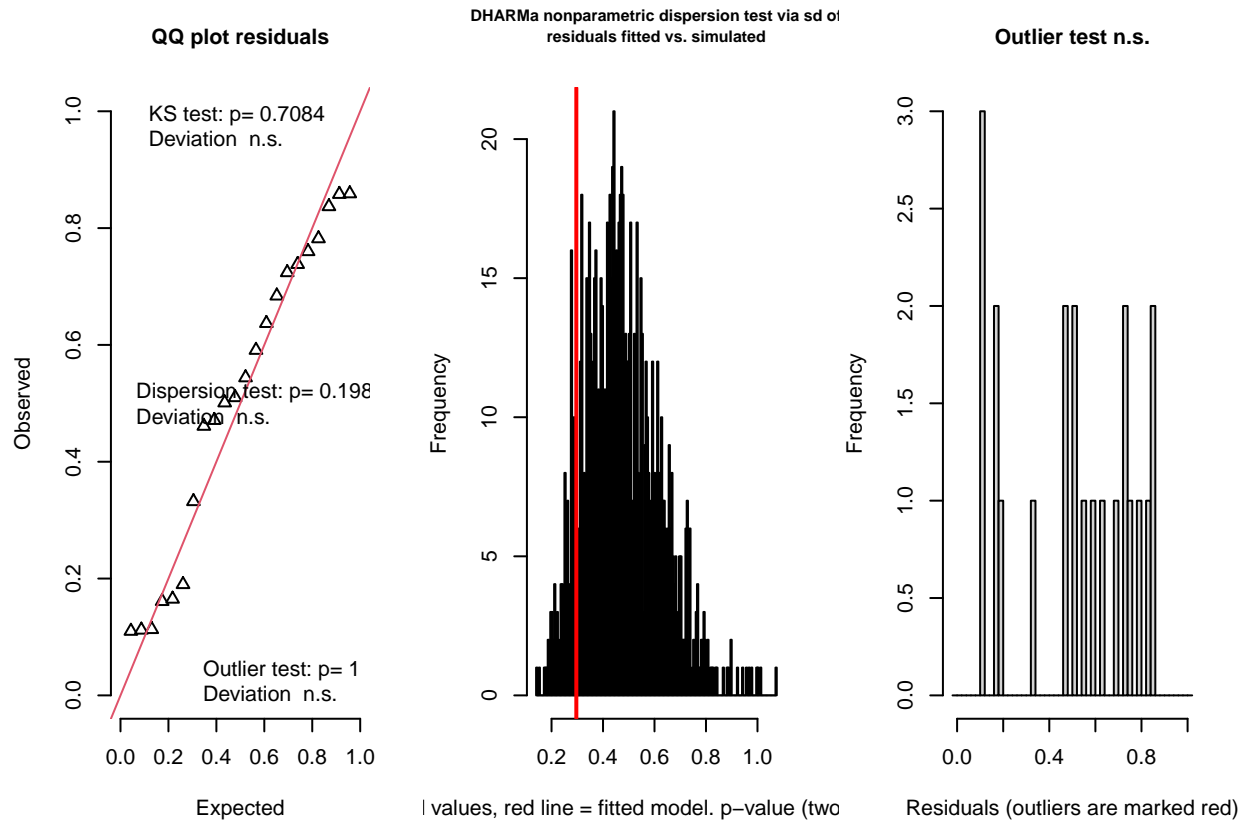
Normal Q-Q Plot



```
trans <- lm(Ab.tuk ~ Treatment + Linage, data = dfWR2nd)
Anova(trans, type="II")
```

```
## Anova Table (Type II tests)
##
## Response: Ab.tuk
##           Sum Sq Df F value    Pr(>F)
## Treatment  0.02523  1   0.4089  0.53363
## Linage      1.26758  7   2.9349  0.04463 *
## Residuals  0.80210 13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_trans <- simulateResiduals(trans, n = 1000)
testResiduals(res_trans)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.14282, p-value = 0.7084
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.61762, p-value = 0.198
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 22, p-value = 1
```

```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1543725
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

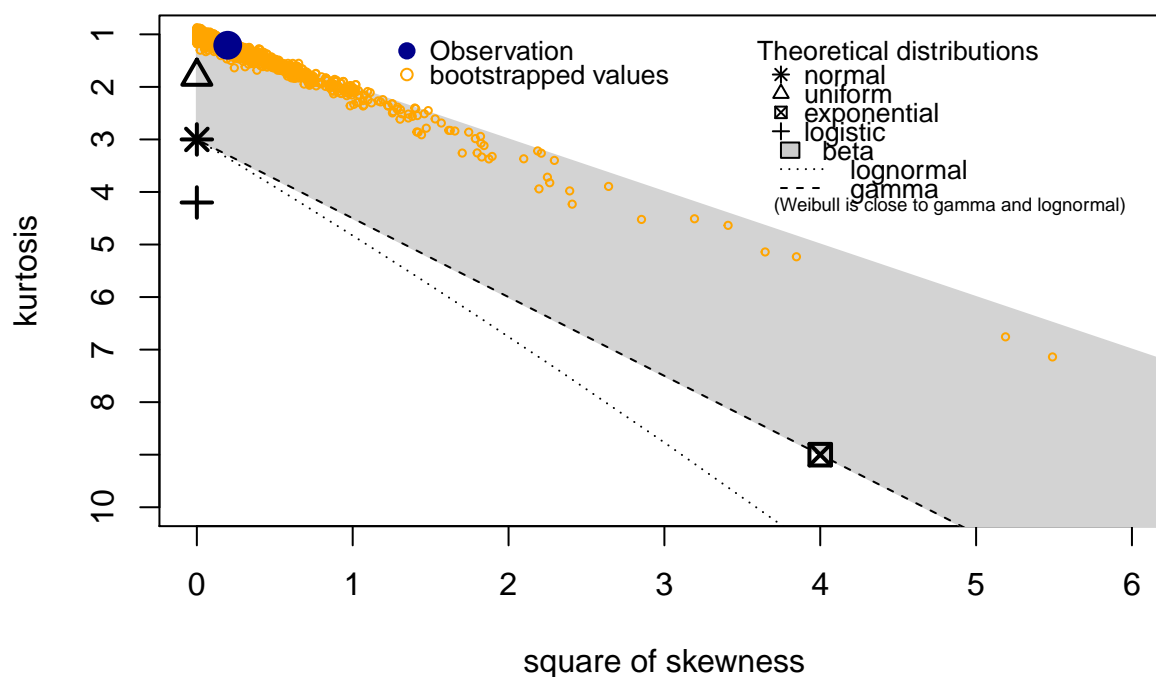
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.14282, p-value = 0.7084
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.61762, p-value = 0.198
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 22, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1543725
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

dataset *Pseudoxanthomonas*

```
descdist(dfPCR$Abundance, boot = 1000)
```

Cullen and Frey graph



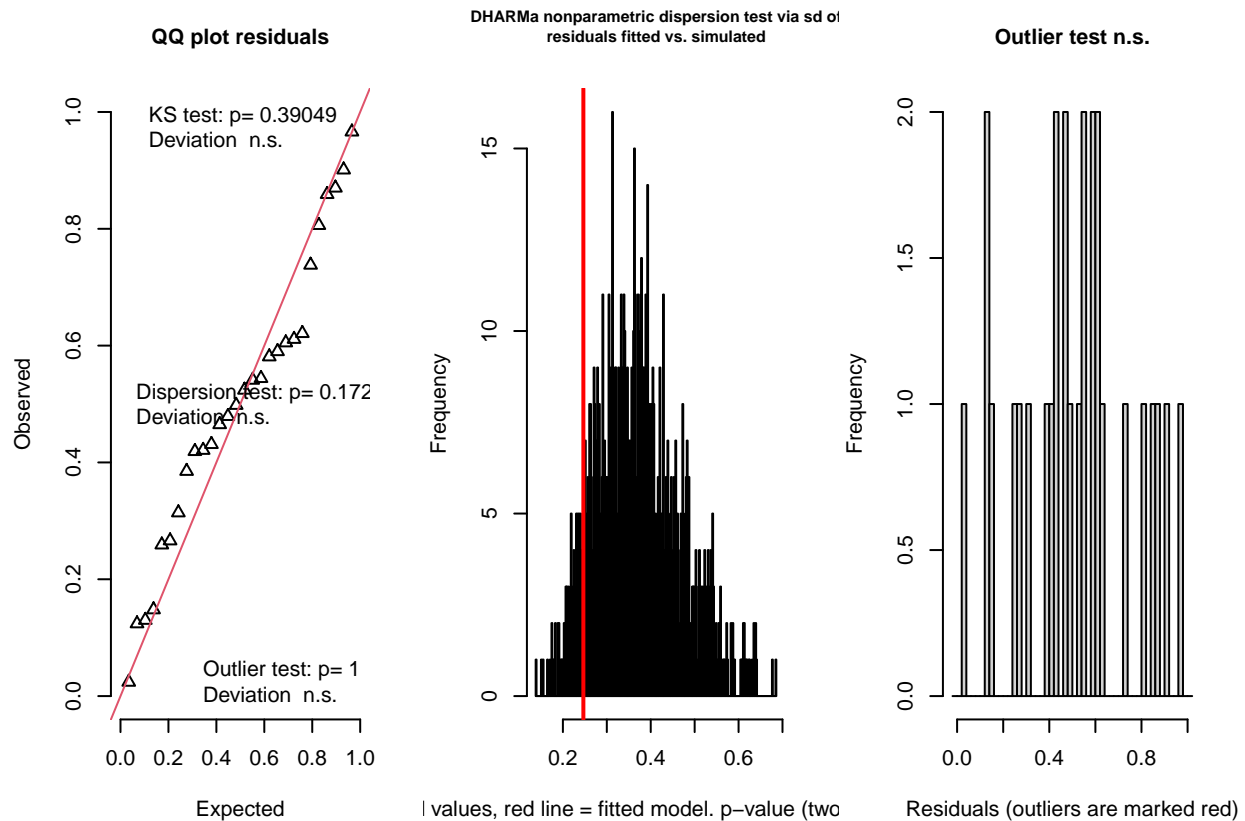
```
## summary statistics
## -----
## min: 0.01552795 max: 0.9994271
## median: 0.9602095
## mean: 0.613549
## estimated sd: 0.4394213
## estimated skewness: -0.4451336
## estimated kurtosis: 1.207382
```

```
logistic <- function(p) log(p / (1-p) +0.01)

dfPCR.mod <- lm(logistic(Abundance) ~ Treatment + Linage, data = dfPCR)
Anova(dfPCR.mod, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(Abundance)
##          Sum Sq Df F value    Pr(>F)
## Treatment   1.116  1  0.1770 0.6789498
## Linage    299.461  8  5.9393 0.0008398 ***
## Residuals  113.445 18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_dfPCR.mod <- simulateResiduals(dfPCR.mod, n = 1000)
testResiduals(res_dfPCR.mod)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.16471, p-value = 0.3905
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.66538, p-value = 0.172
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
```

```

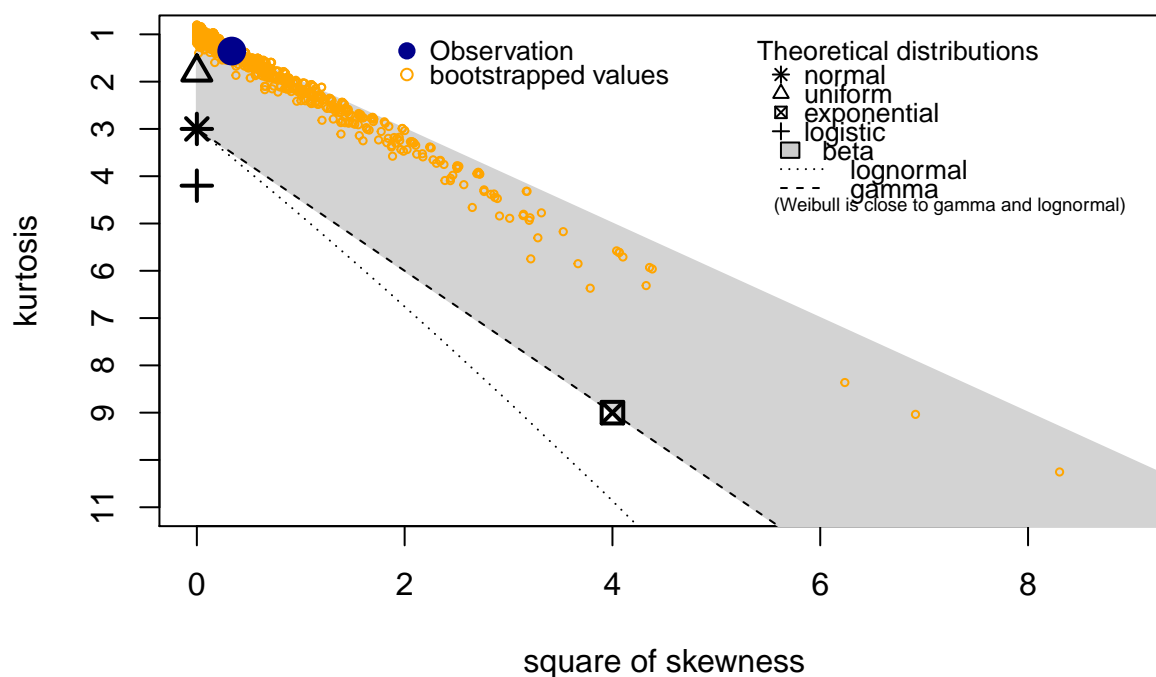
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 28, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1234361
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.16471, p-value = 0.3905
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.66538, p-value = 0.172
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 28, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1234361
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

```
descdist(dfPR2nd$Abundance, boot = 1000)
```

Cullen and Frey graph

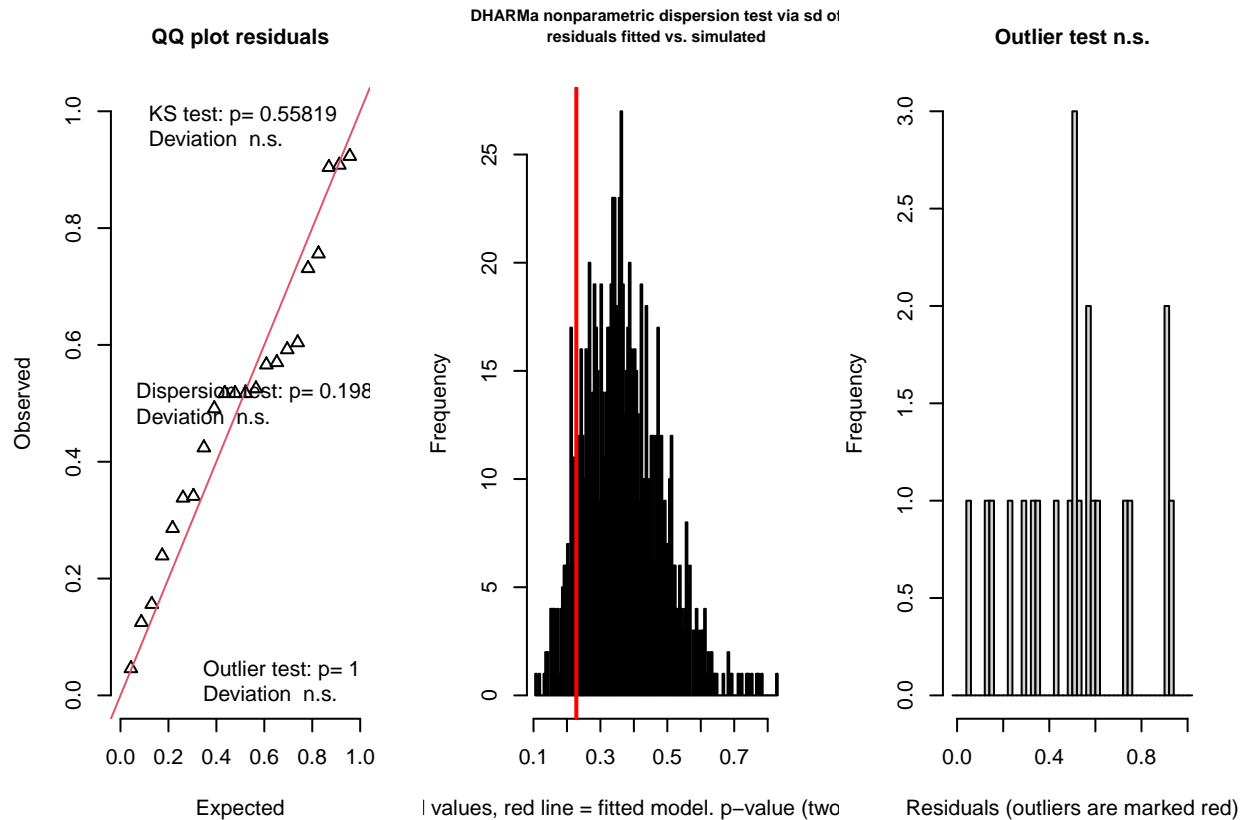


```
## summary statistics
## -----
## min: 0.006869901 max: 0.9993608
## median: 0.9616264
## mean: 0.6327369
## estimated sd: 0.4342354
## estimated skewness: -0.5796202
## estimated kurtosis: 1.353467
```

```
dfPR2nd.mod <- lm(logistic(Abundance) ~ Treatment + Linage, data = dfPR2nd)
Anova(dfPR2nd.mod, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(Abundance)
##      Sum Sq Df F value    Pr(>F)
## Treatment    0.996  1  0.1634 0.692660
## Linage    228.179  7  5.3447 0.004643 **
## Residuals   79.286 13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_dfPR2nd.mod <- simulateResiduals(dfPR2nd.mod, n = 1000)
testResiduals(res_dfPR2nd.mod)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.16873, p-value = 0.5582
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.61762, p-value = 0.198
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 22, p-value = 1
```



```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1543725
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.16873, p-value = 0.5582
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.61762, p-value = 0.198
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 22, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.1543725
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

testing difference of ‘successfull’ vs. ‘failed’ nests

create subset of samples from ‘removal’ group

```

REM <- subset_samples(rel.R2nd, Treatment == "removal")
REM <- prune_taxa(taxa_sums(REM) > 0, REM)

sample_data(REM)

```

```

##           Sample   Nest Linage Treatment age_sampling_d   Group
## B0-19a    B0-19a  B0-19    B0    removal             31 Bacteria
## B0-22a    B0-22a  B0-22    B0    removal             31 Bacteria

```

```
## B15-07a B15-07a B15-07 B15 removal 29 Bacteria
## B16-01a B16-01a B16-01 B16 removal 32 Bacteria
## B16-31a B16-31a B16-31 B16 removal 30 Bacteria
## B20-04a B20-04a B20-04 B20 removal 35 Bacteria
## B20-08a B20-08a B20-08 B20 removal 31 Bacteria
## B24-02a B24-02a B24-02 B24 removal 32 Bacteria
## B24-26a B24-26a B24-26 B24 removal 32 Bacteria
## B24-28a B24-28a B24-28 B24 removal 32 Bacteria
## B36-14a B36-14a B36-14 B36 removal 19 Bacteria
## B36-31a B36-31a B36-31 B36 removal 32 Bacteria
## B39-17a B39-17a B39-17 B39 removal 31 Bacteria
## B7-08a B7-08a B7-08 B7 removal 33 Bacteria
```

```
suc = c("successfull", "successfull", "successfull", "failed", "successfull", "failed", "successfull",
sample_data(REM)$success <- suc

dfREM <- REM %>%
  tax_glom(taxrank = "Genus") %>%
  psmelt()
```

```
otu.REM <- abundances(REM)
meta.REM <- meta(REM)

set.seed(1)
adonis2(distance(REM, method = "bray") ~ success, data = meta.REM)
```

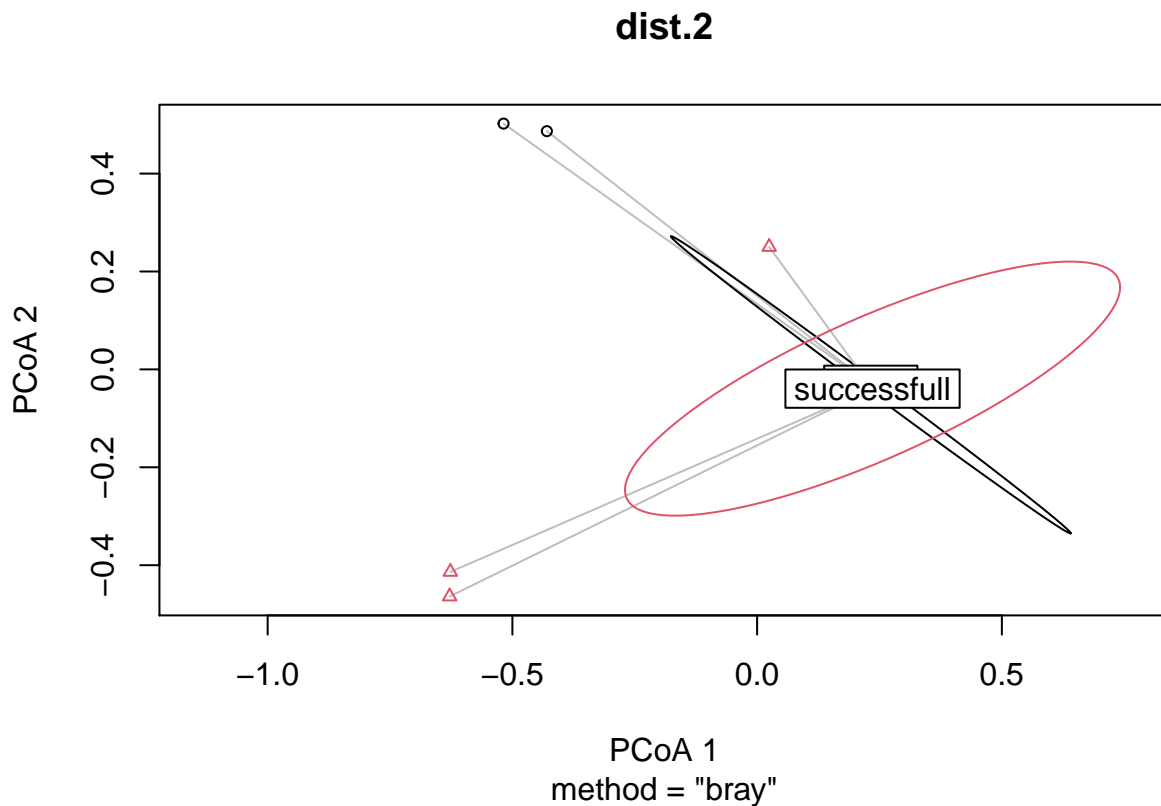
Permanova on data REM

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = distance(REM, method = "bray") ~ success, data = meta.REM)
##           Df SumOfSqs      R2      F Pr(>F)
## success    1  0.19689 0.07083 0.9147  0.573
## Residual  12  2.58296 0.92917
## Total     13  2.77985 1.00000

dist <- vegdist(t(otu.REM))
dist.2 <- betadisper(dist, meta.REM$success)
anova(dist.2)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      1 0.00368 0.003684  0.0205 0.8885
## Residuals  12 2.15427 0.179522
```

```
plot(dist.2, hull = FALSE, ellipse = TRUE)
```

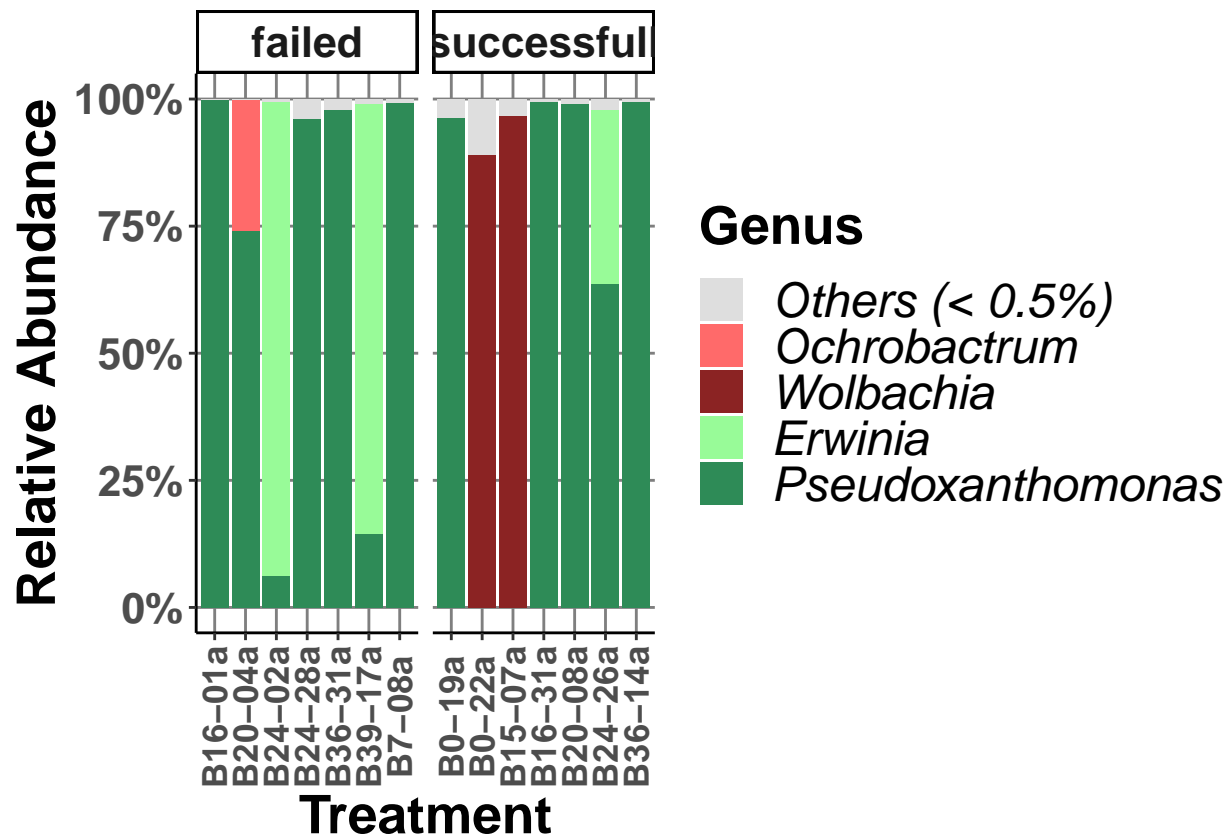


prepare data and plot

```
REM.plot <- dfREM
REM.plot$Success <- factor(REM.plot$success, levels = c("successfull", "failed"))
REM.plot$Genus<-as.character(REM.plot$Genus)
REM.plot$Genus[REM.plot$Abundance<0.05]<-"Others"
REM.plot$Genus<-factor(REM.plot$Genus, levels = c("Others", "Ochrobactrum", "Wolbachia", "Erwinia", "Pseu

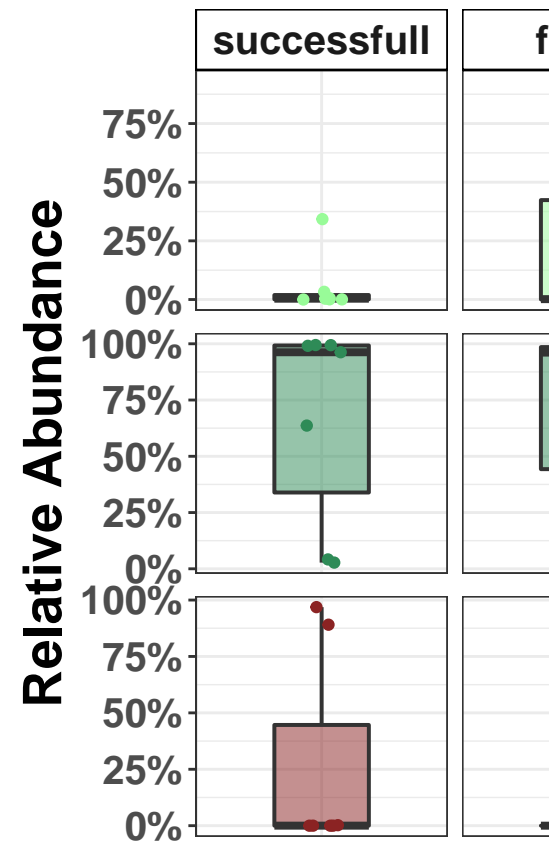
Fungi_Success_plot <-ggplot(REM.plot, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", color = NA, position="fill") +
  scale_fill_manual(values = Plot_colors_g2, name = "Genus") +
  facet_grid(~success, scales = "free_x", space = "free_x")

FS<-Fungi_Success_plot +
  theme(plot.title = element_text(size = 20, face = "bold")) +
  theme_classic()+
  theme(panel.grid.major = element_line(colour = "grey50"))+
  labs(x="Treatment", y="Relative Abundance")+
  scale_y_continuous(labels=percent_format())+
  theme(axis.text.x=element_text(size = 13, angle = 90, vjust = 0.5))+
  theme(text = element_text(size=20, face = "bold"))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 18))+
  theme(legend.text = element_text(face = "italic"))
```



```
dfREM.sub <- subset(dfREM, Genus == "Wolbachia" | Genus == "Pseudoxanthomonas" | Genus == "Erwinia")
ptax_col <- c("palegreen", "seagreen", "brown4")
ptax<-ggplot(data = dfREM.sub, aes(x = success, y = Abundance)) +
  geom_boxplot(aes(fill=Genus),alpha=0.5,lwd=0.7, position = position_dodge(width = 0.3), width=0.45,ou
  scale_fill_manual(values = ptax_col, name = "Genus")+
  scale_color_manual(values = ptax_col, name = "Genus")+
  labs(x = "", y = "Abundance\n")+
  facet_grid(Genus~fct_relevel(success, "successfull", "failed"), scales = "free")+theme_bw()
ptax<-ptax+ theme(legend.position="right")+ylab("Relative Abundance")
ptax<-ptax+ theme(legend.text=element_text(size=15, face = "italic"))+
  theme(legend.key = element_rect(color = NA, fill = NA),legend.key.size = unit(0.9, "cm"))+
  theme(legend.title = element_text(size = 20, face = "bold"))+
  scale_y_continuous(labels=percent_format())
abu2<-ptax + theme(strip.background =element_rect(fill="white", color="black"))+
  theme(strip.text.x = element_text(size = 15, face = "bold"))
Success.plot<-abu2+theme(axis.title.y = element_text(size=20, face="bold"))+theme(axis.text.y = element
  theme(axis.text.x = element_text(size=20, angle=90,face="bold"))+
  theme(axis.title.x = element_blank(), axis.text.x=element_blank(),
    axis.ticks.x=element_blank())+
  theme(strip.text.y = element_blank())
```

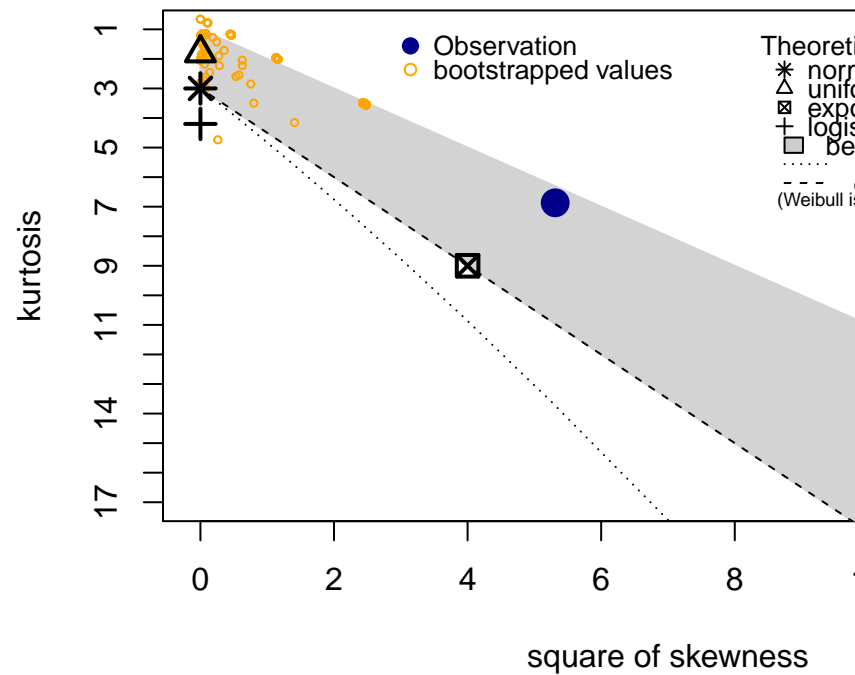
```
Success.plot_bac <- Success.plot + theme(panel.spacing.y = unit(0.3, "cm"))
Success.plot_bac
```



plot core taxa of successfull vs. failed nests with relative abundance

```
Wrem <- subset(dfREM, Genus == "Wolbachia")
descdist(Wrem$Abundance, boot = 1000)
```

Cullen and Frey graph



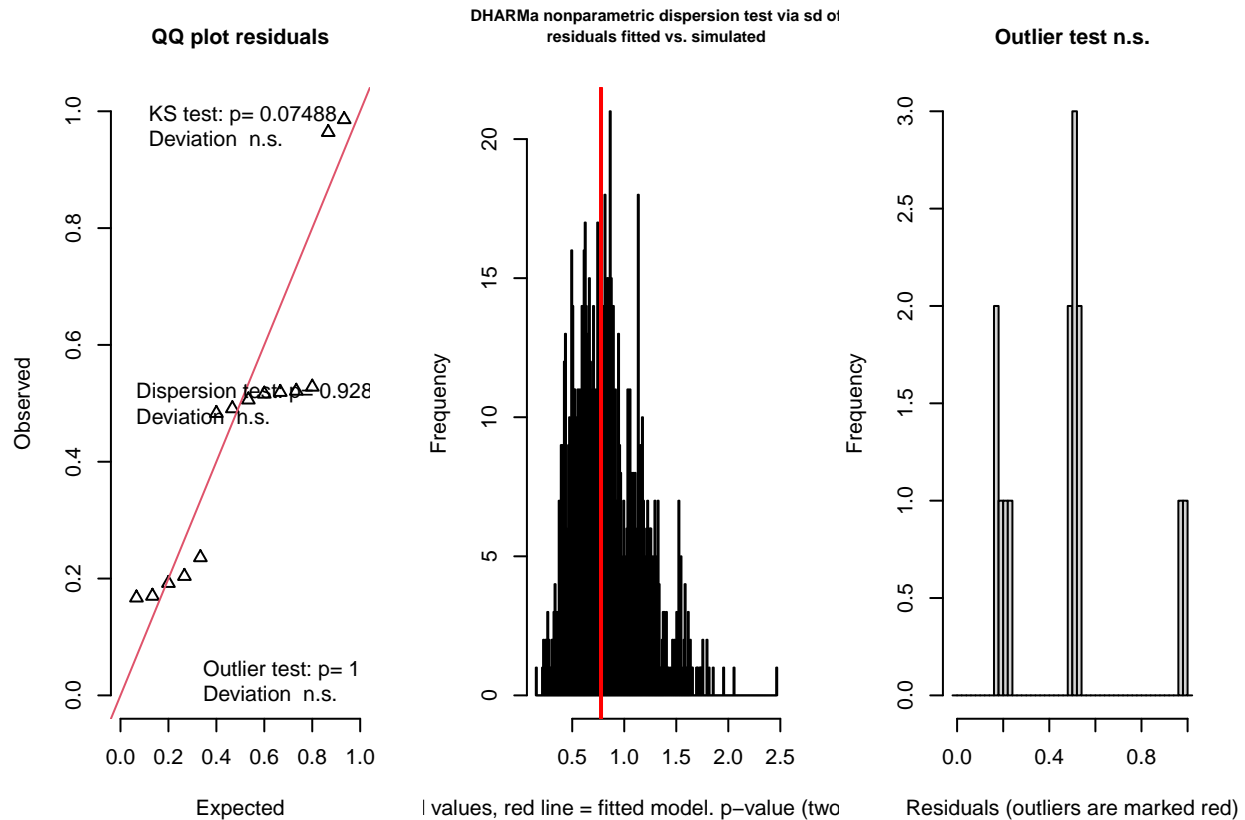
test if taxa are different in these two groups

```
## summary statistics
## -----
## min: 0    max: 0.9685696
## median: 6.920423e-05
## mean: 0.1330016
## estimated sd: 0.337838
## estimated skewness: 2.304651
## estimated kurtosis: 6.869043
```

```
Wrem.mod <- lm(logistic(Abundance) ~ success, data = Wrem)
Anova(Wrem.mod, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(Abundance)
##           Sum Sq Df F value Pr(>F)
## success   15.912  1  2.4623 0.1426
## Residuals  77.548 12
```

```
res_Wrem.mod <- simulateResiduals(Wrem.mod, n = 1000)
testResiduals(res_Wrem.mod)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.32914, p-value = 0.07488
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMa outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
```

```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.32914, p-value = 0.07488
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

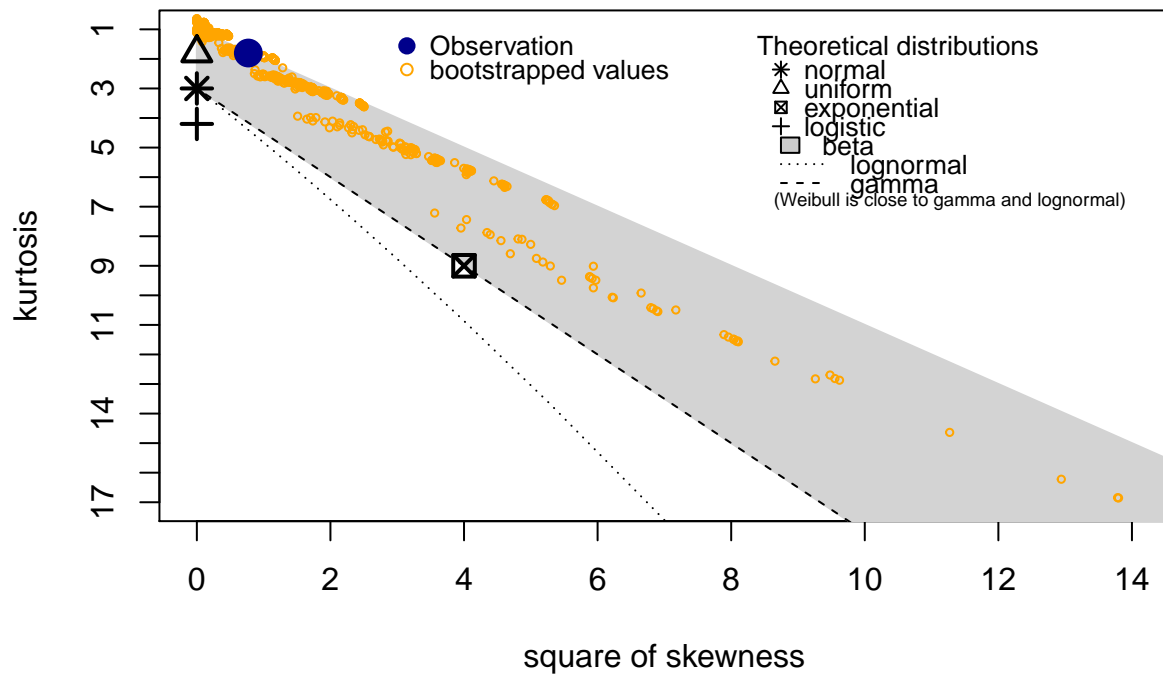
```

```

Prem <- subset(dfREM, Genus == "Pseudoxanthomonas")
descdist(Prem$Abundance, boot = 1000)

```


Cullen and Frey graph

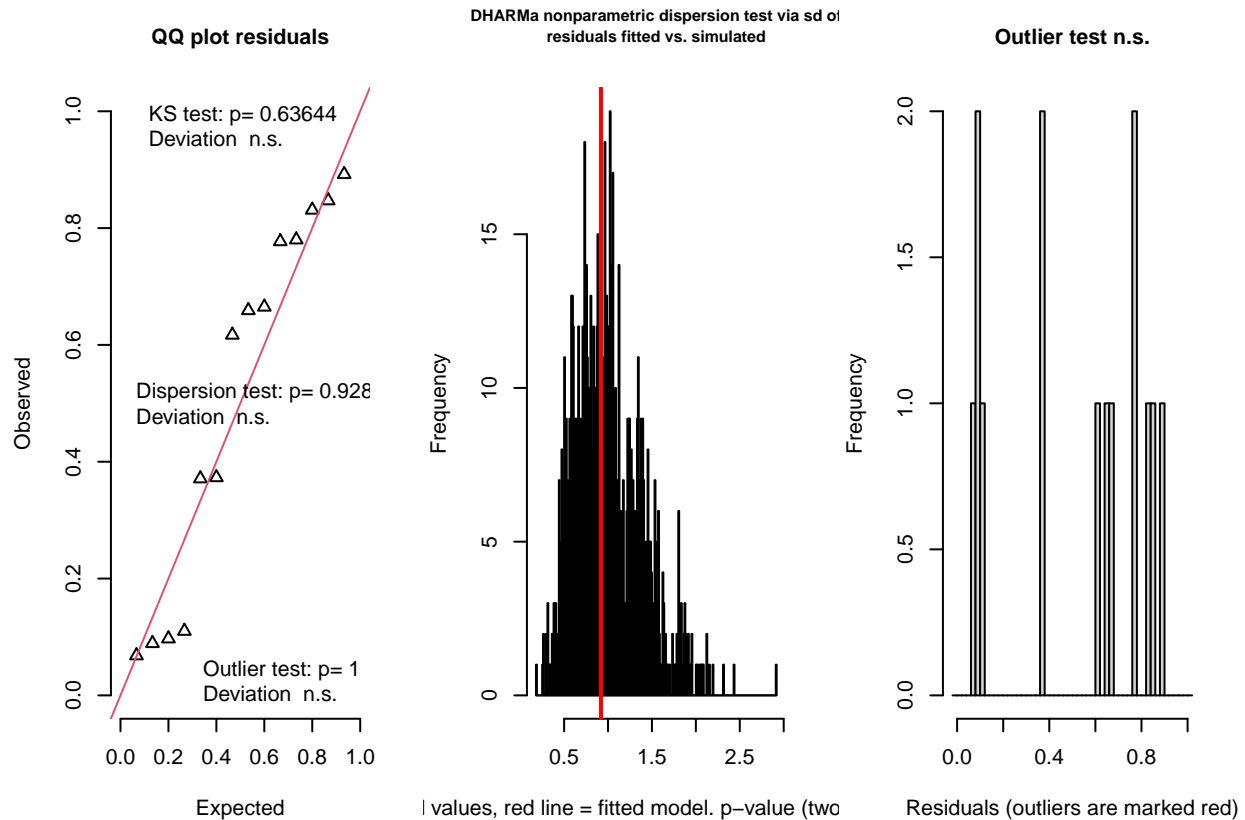


```
## summary statistics
## -----
## min: 0.02758178   max: 0.9987783
## median: 0.9616264
## mean: 0.680395
## estimated sd: 0.4157842
## estimated skewness: -0.8779036
## estimated kurtosis: 1.800568
```

```
Prem.mod <- lm(logistic(Abundance) ~ success, data = Prem)
Anova(Prem.mod, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(Abundance)
##           Sum Sq Df F value Pr(>F)
## success    0.562  1  0.0441 0.8371
## Residuals 152.746 12
```

```
res_Prem.mod <- simulateResiduals(Prem.mod, n = 1000)
testResiduals(res_Prem.mod)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.18843, p-value = 0.6364
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
```

```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.18843, p-value = 0.6364
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

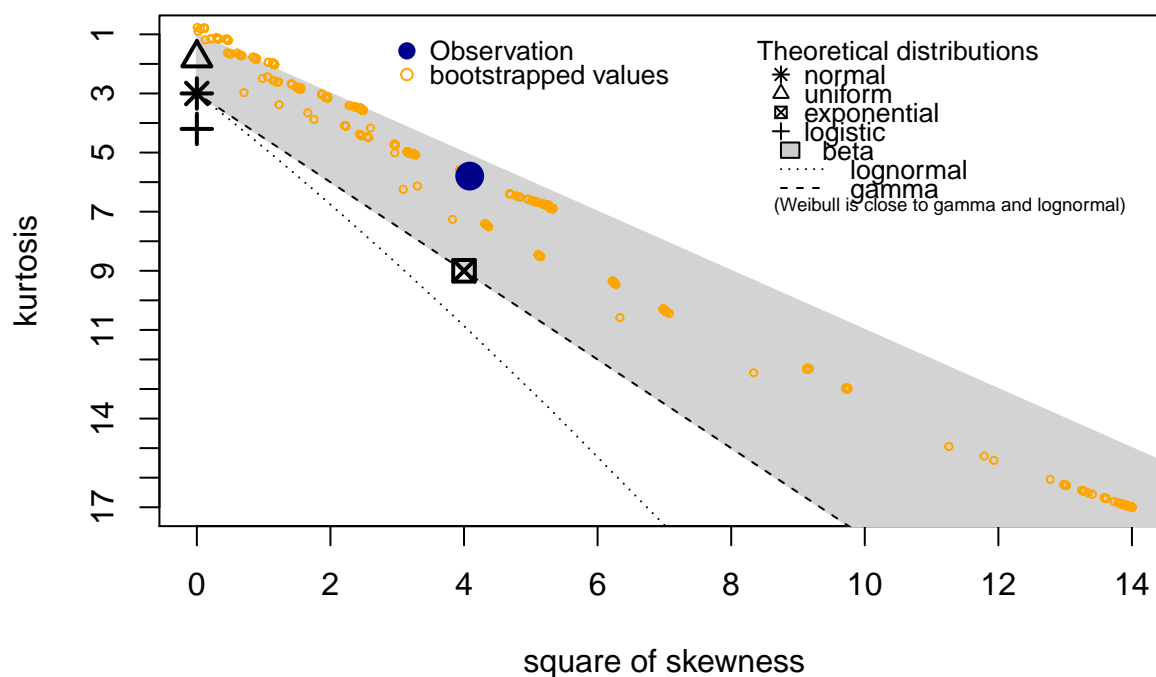
```

```

Erem <- subset(dfREM, Genus == "Erwinia")
descdist(Erem$Abundance, boot = 1000)

```

Cullen and Frey graph

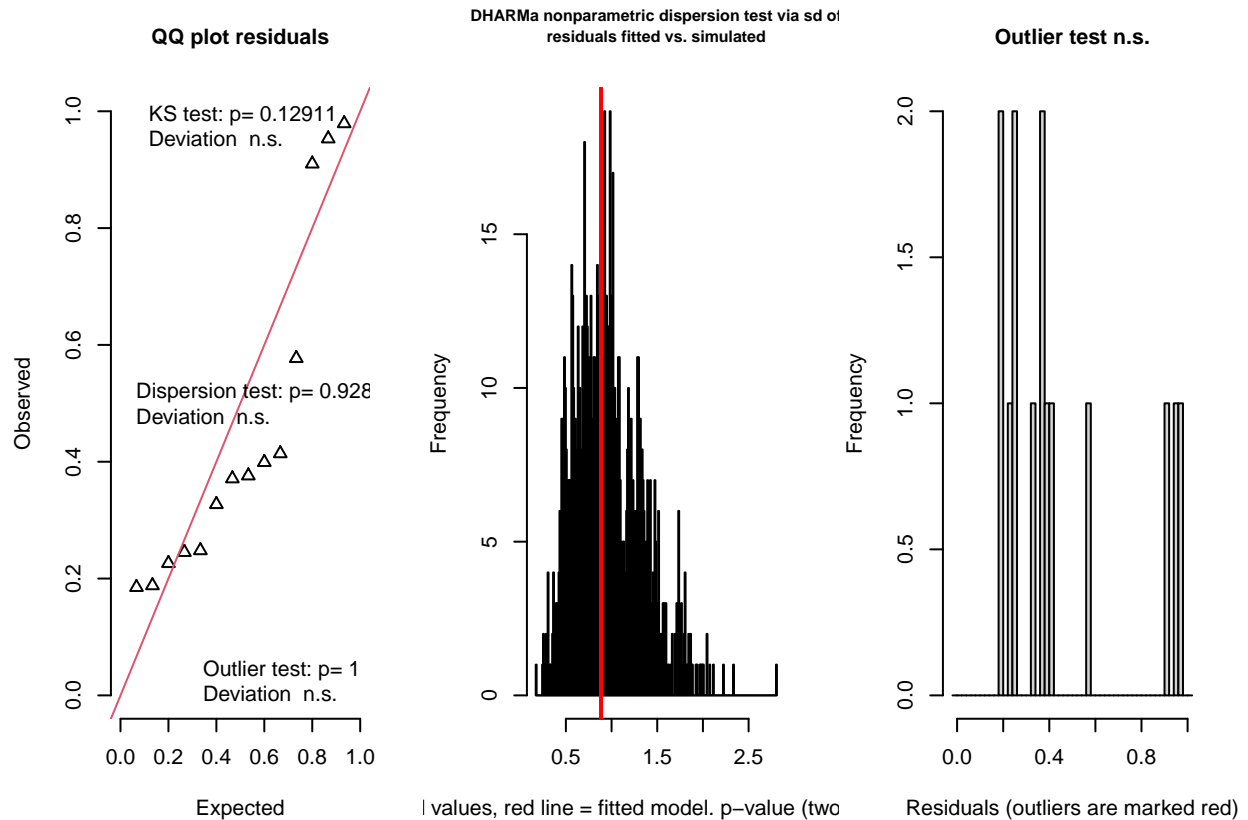


```
## summary statistics
## -----
## min: 0    max: 0.9314597
## median: 0.001771372
## mean: 0.154791
## estimated sd: 0.3240921
## estimated skewness: 2.021202
## estimated kurtosis: 5.796488
```

```
Erem.mod <- lm(logistic(Abundance) ~ success, data = Erem)
Anova(Erem.mod, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: logistic(Abundance)
##          Sum Sq Df F value Pr(>F)
## success    4.098  1  0.6406  0.439
## Residuals 76.756 12
```

```
res_Erem.mod <- simulateResiduals(Erem.mod, n = 1000)
testResiduals(res_Erem.mod)
```



```
## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.30029, p-value = 0.1291
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
```

```

## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

## $uniformity
##
## One-sample Kolmogorov-Smirnov test
##
## data: simulationOutput$scaledResiduals
## D = 0.30029, p-value = 0.1291
## alternative hypothesis: two-sided
##
##
## $dispersion
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.92226, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data: simulationOutput
## outliers at both margin(s) = 0, observations = 14, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.001998002
## 95 percent confidence interval:
## 0.0000000 0.2316358
## sample estimates:
## frequency of outliers (expected: 0.001998001998002 )
## 0

```

References

- Davis NM, Proctor DiM, Holmes SP, Relman DA, Callahan BJ. 2018. Simple statistical identification and removal of contaminant sequences in marker-gene and metagenomics data. *Microbiome*. 6(1). doi:10.1186/S40168-018-0605-2.
- Delignette-Muller ML, Dutang C. 2015. fitdistrplus: An R package for fitting distributions. *J Stat Softw*. 64(4):1–34. doi:10.18637/jss.v064.i04.
- Dimitris Rizopoulos (2022). GLMMadaptive: Generalized Linear Mixed Models using Adaptive Gaussian Quadrature. R package version 0.8-5. <https://CRAN.R-project.org/package=GLMMadaptive>
- Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1-48. doi:10.18637/jss.v067.i01.

John Fox and Sanford Weisberg (2019). An {R} Companion to Applied Regression, Third Edition. Thousand Oaks CA: Sage. URL: <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>

Gavin L. Simpson (2022). permute: Functions for Generating Restricted Permutations of Data. R package version 0.9-7. <https://CRAN.R-project.org/package=permute>

Hartig F. 2021. DHARMA: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models. <https://cran.r-project.org/package=DHARMA>.

Højsgaard, Søren and Halekoh, Ulrich (2022). doBy: Groupwise Statistics, LSmeans, Linear Estimates, Utilities. R package version 4.6.12. <https://CRAN.R-project.org/package=doBy>

Kassambara, Alboukadel (2020). ggpubr: ‘ggplot2’ Based Publication Ready Plots. R package version 0.4.0. <https://CRAN.R-project.org/package=ggpubr>

Kuznetsova A, Brockhoff PB, Christensen RHB (2017). “lmerTest Package: Tests in Linear Mixed Effects Models.” *Journal of Statistical Software*, 82(13), 1-26. doi: 10.18637/jss.v082.i13 (URL: <https://doi.org/10.18637/jss.v082.i13>).

Lahti, Leo et al. microbiome R package. URL: <http://microbiome.github.io>

Lüdtke D (2018). “ggeffects: Tidy Data Frames of Marginal Effects from Regression Models.” *Journal of Open Source Software*, 3(26), 772. doi: 10.21105/joss.00772 (URL: <https://doi.org/10.21105/joss.00772>).

Lüdtke D (2021). *sjPlot: Data Visualization for Statistics in Social Science*. R package version 2.8.10, <URL: <https://CRAN.R-project.org/package=sjPlot>>.

Lüdtke D, Ben-Shachar MS, Patil I, Waggoner P, Makowski D. (2021) performance: An R Package for Assessment, Comparison and Testing of Statistical Models. *Journal of Open Source Software*, 6(60), 3139. doi:10.21105/joss.03139.

Lüdtke et al., (2021). see: An R Package for Visualizing Statistical Models. *Journal of Open Source Software*, 6(64), 3393. <https://doi.org/10.21105/joss.03393>

Mangiafico S. 2021. rcompanion: Functions to Support Extension Education Program Evaluation. <https://cran.r-project.org/package=rcompanion>.

Martinez Arbizu, Pedro (2017). pairwiseAdonis: Pairwise Multilevel Comparison using Adonis. R package version 0.4.

McMurdie PJ, Holmes S. 2013. Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. *PLoS One*. 8(4):e61217. doi:10.1371/journal.pone.0061217.

Mollie E. Brooks, Kasper Kristensen, Koen J. van Benthem, Arni Magnusson, Casper W. Berg, Anders Nielsen, Hans J. Skaug, Martin Maechler and Benjamin M. Bolker (2017). glmmTMB Balances Speed and Flexibility Among Packages for Zero-inflated Generalized Linear Mixed Modeling. *The R Journal*, 9(2), 378-400.

Oksanen J, Blanchet FG, Friendly M, Kindt R, Legendre P, McGlinn D, Minchin PR, O’Hara RB, Simpson GL, Solymos P, et al. 2020a. vegan: Community Ecology Package. <https://cran.r-project.org/package=vegan>.

Pedersen, Thomas Lin (2020). patchwork: The Composer of Plots. R package version 1.1.1. <https://CRAN.R-project.org/package=patchwork>

R Core Team. 2021. R: A language and environment for statistical computing. <https://www.r-project.org/>.

Russell V. Lenth (2022). emmeans: Estimated Marginal Means, aka Least-Squares Means. R package version 1.7.2. <https://CRAN.R-project.org/package=emmeans>

Sarkar, Deepayan (2008) *Lattice: Multivariate Data Visualization with R*. Springer, New York. ISBN 978-0-387-75968-5

Shetty, Sudarshan and Lahti, Leo (2021). microbiomeutilities: Utilities for Microbiome Analytics. R package version 1.00.16.

Slowikowski, Kamil (2021). `ggrepel`: Automatically Position Non-Overlapping Text Labels with ‘`ggplot2`’. R package version 0.9.1. <https://CRAN.R-project.org/package=ggrepel>

Wickham H. 2016. `ggplot2`: Elegant Graphics for Data Analysis. New York: Springer-Verlag New York.

Wickham et al., (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686, <https://doi.org/10.21105/joss.01686>

Wickham H, Seidel D. 2020. `scales`: Scale Functions for Visualization. <https://cran.r-project.org/package=scales>.

Wickham H, François R, Henry L, Müller K. 2021. `dplyr`: A Grammar of Data Manipulation. <https://cran.r-project.org/package=dplyr>.

Hadley Wickham (2021). `forcats`: Tools for Working with Categorical Variables (Factors). R package version 0.5.1. <https://CRAN.R-project.org/package=forcats>

Wilke, Claus O. (2020). `cowplot`: Streamlined Plot Theme and Plot Annotations for ‘`ggplot2`’. R package version 1.1.1. <https://CRAN.R-project.org/package=cowplot>

Yihui Xie (2021). `knitr`: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.37.