Two Common Non-lethal Methods for the Study of the Gut Bacterial Communities in Wild Lizards

Stephanie Hereira-Pacheco, Centro Tlaxcala de Biología de la Conducta, UATx
Mauricio Hernández, Doctorado en CB, Centro Tlaxcala de Biología de la Conducta, UATx

25 - 04 - 2023

Contents

Qiime2 Scripts-16S

Step 1: EXTRACT BARCODES

For this step, It will be used the 'extract_barcodes.py' script used in qiime1.

```
#I'll use one library called "Ste1" with Ste1_1.fastq and Ste2_1.fastq

extract_barcodes.py  -f Sg_16S-5_1.fastq -r Sg_16S-5_2.fastq --bc1_len 8
--bc2_len 8 -c barcode_paired_end -o library5_extract_barcodes

extract_barcodes.py  -f Sg_16S-6_1.fastq -r Sg_16S-6_2.fastq --bc1_len 8
--bc2_len 8 -c barcode_paired_end -o library6_extract_barcodes

extract_barcodes.py  -f Sg_16S-7_1.fastq -r Sg_16S-7_2.fastq --bc1_len 8
--bc2_len 8 -c barcode_paired_end -o library7_extract_barcodes
```

-f : forward reads

-r : reverse reads

-c: input type [default: barcode_single_end]

__-bc1_len and –bc2_len : Specify the length, in base pairs, of barcodes

-o : output

Step 2: IMPORT TO QIIME AND DEMULTIPLEX SEQUENCES

For this step, we need to create a directory with the three files output from the previous step, containing:

1. forward.fastq.gz: file that contains the forward sequence reads
2. reverse.fastq.gz: file that contains the reverse sequence reads
3. barcdes.fastq.gz: file that contains the barcode sequence reads

```
qiime tools import --type EMPPairedEndSequences
--input-path library5_extract_barcodes/
--output-path L5.qza

qiime tools import --type EMPPairedEndSequences
--input-path library6_extract_barcodes/
--output-path L6.qza

qiime tools import --type EMPPairedEndSequences
--input-path library7_extract_barcodes/
--output-path L7.qza
```

–type : type of file , in this case paired end sequences. Check other import types[1].

–input-path: directory with the files to import

–output-path: artifact name output

And then, we perform the demultiplexing:

```
qiime  demux emp-paired  --i-seqs L5.qza
--m-barcodes-file Library5_SgHC_and_SgExtra.txt
--m-barcodes-column barcode-sequence --output-dir demux_L5
--p-no-golay-error-correction

qiime  demux emp-paired  --i-seqs L6.qza
--m-barcodes-file Library6_SgHC_and_SgExtra.txt
--m-barcodes-column barcode-sequence
--output-dir demux_L6 --p-no-golay-error-correction
```

[1] https://docs.qiime2.org/2021.4/tutorials/importing/

```
qiime  demux emp-paired  --i-seqs L7.qza
--m-barcodes-file Library7_Sg_DigestiveTract.txt
--m-barcodes-column BarcodeSequence
--output-dir demux_L7 --p-no-golay-error-correction
```

–i-seqs : artifact with the import paired end sequences

–m-barcodes-file : mapping file containing information of the sequences

–m-barcodes-column: column name of the Barcode sequences

–output-dir : output directory with the demultiplexed samples and error correction details

–p-no-golay-error-correction: by default perform a correction with a barcode of 12 nt if not use this option (in our case is 16 nt)

Step 3: REMOVE PRIMERS AND VISUALIZATION

```
qiime cutadapt trim-paired
--i-demultiplexed-sequences demux_L5/per_sample_sequences.qza
--p-front-f CCTACGGGNGGCWGCAG
--p-front-r GACTACHVGGGTATCTAATCC
--o-trimmed-sequences demux_L5/per_sample_sequences_trimmed.qza

qiime cutadapt trim-paired
--i-demultiplexed-sequences demux_L6/per_sample_sequences.qza
--p-front-f CCTACGGGNGGCWGCAG
--p-front-r GACTACHVGGGTATCTAATCC
--o-trimmed-sequences demux_L6/per_sample_sequences_trimmed.qza

qiime cutadapt trim-paired
--i-demultiplexed-sequences demux_L7/per_sample_sequences.qza
--p-front-f CCTACGGGNGGCWGCAG --p-front-r GACTACHVGGGTATCTAATCC
--o-trimmed-sequences demux_L7/per_sample_sequences_trimmed.qza
```

–i-demultiplexed-sequences : demultiplexed sequences (.qza artifact)

–p-cores : number of threads

–p-front-f : forward primer sequences (front if is in the beginning of the sequences)

–p-front-r : reverse primer sequences (front if is in the beginning of the sequences)

–o-trimmed-sequences : output

```
qiime demux summarize
--i-data demux_L5/per_sample_sequences_trimmed.qza
--o-visualization  trimmed_l5.qzv


qiime demux summarize
--i-data demux_L6/per_sample_sequences_trimmed.qza
--o-visualization  trimmed_l6.qzv

qiime demux summarize
--i-data demux_L7/per_sample_sequences_trimmed.qza
--o-visualization  trimmed_l7.qzv
```

–i-data : demultiplexed and/or trimmed sequences

–o-visualization : output

In this case, due to de the low quality of reverse reads we will continue with the forward and reverse sequences and let's set the truncation length of 260 bp for forward and 200 bp for reverse.

Step 4: RUN DADA2

In this step, we will perform as an example a loop that can be used in the previous steps and the next ones:

```
qiime dada2 denoise-paired
--i-demultiplexed-seqs demux_L5/per_sample_sequences_trimmed.qza
--p-trunc-len-f 260 --p-trunc-len-r 200 --output-dir dada2_l5_paired

qiime dada2 denoise-paired
--i-demultiplexed-seqs demux_L6/per_sample_sequences_trimmed.qza
--p-trunc-len-f 260 --p-trunc-len-r 200 --output-dir dada2_l6_paired

qiime dada2 denoise-paired
--i-demultiplexed-seqs demux_L7/per_sample_sequences_trimmed.qza
--p-trunc-len-f 260 --p-trunc-len-r 200 --output-dir dada2_l7_paired
```

–i-demultiplexed-seqs : demultiplexed and trimmed sequences

-p-trunc-len-f : length to trunc in forward sequences sequences to obtain good quality (usually when sequencing drops)

-p-trunc-len-r : length to trunc in resverse sequences sequences to obtain good quality (usually when sequencing drops)

–output-dir : output directory that will contain feature-table and representative sequences

In case we want to visualize the results from dada2 (table, seqs and stats):

```
#example using dada2_l5_paired (sample)
cd dada2_l5_paired

qiime  metadata tabulate
--m-input-file denoising_stats.qza
--o-visualization denoising_stats_paired.qzv

qiime  metadata tabulate
--m-input-file representative_sequences.qza
--o-visualization representative_sequences.qzv

qiime feature-table summarize
--i-table table.qza --o-visualization table.qzv
```

–m-input-file : stats or sequences

–i-table : table

–o-visualization: output

Step 5: MERGING TABLES AND SEQUENCES

First, merge tables and seqs:

```
qiime feature-table merge
--i-tables dada2_l5_paired/table.qza
--i-tables dada2_l6_paired/table.qza
--i-tables dada2_l7_paired/table.qza
--o-merged-table merge_table.qza
```

–i-tables : table to merge (put every time you want to add a different table)

–o-merged-table : output/merge table

```
qiime feature-table merge-seqs \
--i-data dada2_l5_paired/representative_sequences.qza \
--i-data dada2_l6_paired/representative_sequences.qza \
--i-data dada2_l7_paired/representative_sequences.qza \
--o-merged-data merge_seqs.qza
```

–i-data : sequences to merge (put every time you want to add a different sequence)

–o-merged-data : output/merge sequences

Then, let's visualize them:

```
qiime feature-table summarize \
--i-table merge_table.qza\
--m-sample-metadata-file mapping_file.txt
--o-visualization merge_table.qzv \
```

–i-table : merged table

–m-sample-metadata-file : mapping file containing all libraries

–o-visualization : output/ visualization artifact

```
qiime metadata tabulate  \
--m-input-file merge_seqs_dada.qza \
--o-visualization merge_seqs.qzv\
```

–m-input-file : merged sequences

–o-visualization : output/ visualization artifact

Step 6: ASSIGN TAXONOMY

```
qiime feature-classifier classify-sklearn
--i-reads merge_seqs.qza
--i-classifier /home/steph/Downloads/gg-13-8-99-nb-classifier.qza
--o-classification taxonomy.qza
```

cclassify-sklearn : using sklearn (other options are vsearch and blast)

–i-reads : seqs merged

–i-rclassifier: artifact classifier full-length (https://docs.qiime2.org/2021.4/data-resources/)

–o-classification output artifact with taxonomy

Step 7: FILTERING TABLE

- **Removing taxa of chloroplast and mitochondria**

  I checked the feature table and the division Phragmoplastophyta is all assigned to plants

```
qiime taxa filter-table
--i-table merge_table.qza
--i-taxonomy taxonomy.qza
--p-exclude mitochondria,chloroplast
--o-filtered-table merge_table_filtered.qza
```

–i-table : merge table

–i-taxonomy : taxonomy (from assign taxonomy)

–p-exclude : taxa to exclude

–o-filtered-table : output/artifact

- **Visualizing the taxonomy in a barplot**

```
qiime  taxa  barplot
--i-table merge_table_filtered.qza
--i-taxonomy taxonomy.qza
--m-metadata-file mapping_file.txt
--o-visualization barplot_filtered.qzv

qiime tools view barplot_filtered.qzv
```

–i-table : input table

–m-metadata-file : mapping file

–i-taxonomy : taxonomy

–o-visualization: .qzv of barplot

Step 8: FILTERING SEQUENCES

For this step we will filter the representative sequences base on the table filtered.

```
qiime feature-table filter-seqs
--i-data merge_seqs.qza
--i-table merge_table_filtered.qza
--o-filtered-data merge_seqs_filtered.qza
```

–i-data : input sequences

–i-table : input table use to filter

–o-filtered-data : output/filtered sequences

Step 9: BUILDING THE TREE

For this step we will build the phylogenetic tree *denovo.*

```
qiime phylogeny align-to-tree-mafft-fasttree
--i-sequences merge_seqs_filtered.qza
--output-dir phylo_tree
```

–i-sequences : sequences filtered

–output-dir : output director that will contain the alignment, masked alignment, the tree and the rooted treed.

Step 10: EXPORTING TABLE AND TAXONOMY TO OTUTABLE

```
#export feature-table
qiime tools export --input-path merge_table_filtered.qza --output-path feature-table

#export taxonomy
qiime tools export --input-path taxonomy.qza --output-path feature-table

#site in feature-table directory
cd feature-table/

#before this change the headers from taxonomy.tsv (Fearure.ID= #OTUID, Taxa=taxonomy)

#add taxonomy to biom-table
biom add-metadata -i feature-table.biom
--observation-metadata-fp taxonomy.tsv -o feature-table-taxonomy.biom

#convert biom to tsv to check the otutable
biom convert -i feature-table-taxonomy.biom
-o feature-table-taxonomy.txt --to-tsv --header-key taxonomy
```

–input-path: artifact to export (table or taxonomy)

–output-path: directory outpur

-i : feature-table in biom format

–observation-metadata-fp : taxonomy file (already changed)

-o: output

–to-tsv –header-key taxonomy : options to convert and add taxonomy to otutable

Barplots Phylum and Genera

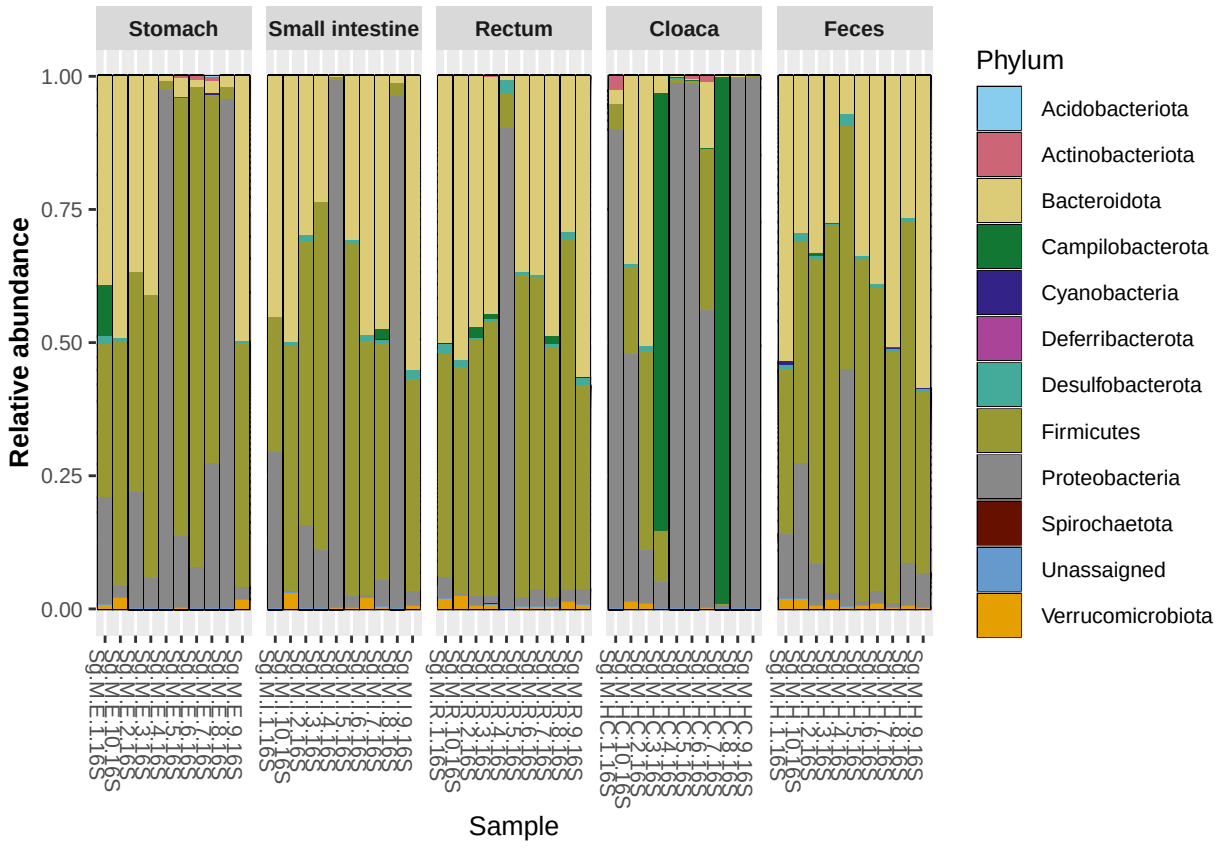
Phylum

```
## Loading libraries
library(phyloseq)
library(ggplot2)
library(vegan)
library(picante)
library(ape)
library(devtools)
library(vegan)
library("RColorBrewer")
library(scales)
library(grid)
library(reshape2)
library(dplyr)
library(scales)
library(viridis)
library(hrbrthemes)
library(gcookbook)
library(tidyverse)
library(dplyr)
```

```r
# Nota: merge_phyloseq combina la información de todas las tablas.

metadata <- read.csv(file = "../Data/metadata.csv", header = TRUE, row.names = 1) %>% dplyr::select(Ind
otu_table <- read.csv("../Data/feature_table.csv", header = TRUE, row.names = 1)
taxonomy <- read.csv("../Data/taxonomy.csv", header = TRUE, row.names = 1) %>%
  mutate_at(c("Phylum"), str_replace,"p__", "")
#phylo<-read.tree(file = "tree.nwk")

# Crear objeto de categoría phyloseq
SAM <- sample_data(metadata)
TAX <- tax_table(as.matrix(taxonomy))
OTU <- otu_table(otu_table, taxa_are_rows = TRUE)
#PHY<-phy_tree(phylo)
physeq <- merge_phyloseq(OTU, TAX, SAM)



# Convertir los recuentos de OTUs en abundancias relativas y normalizar
# la abundacia de cada OTU
relative  = transform_sample_counts(physeq = physeq, function(OTU) OTU / sum(OTU))

# Remover las bacterias sin identificacion a nivel Kingdom
physeq_sub1 <- subset_taxa(physeq, !is.na(Kingdom) & !Kingdom %in% c("", "Unassigned"))
physeq_sub2 <- subset_taxa(physeq, !is.na(Genus) & !Genus %in% c("", "Unassigned"))

##############################################################################
# Abundancia relativa por muestra a nivel de phylum (Original paper)
# Genera paleta de colores
library(rcartocolor)
my_colors = carto_pal(12, "Safe")
my_colors
```

```
##  [1] "#88CCEE" "#CC6677" "#DDCC77" "#117733" "#332288" "#AA4499" "#44AA99"
##  [8] "#999933" "#882255" "#661100" "#6699CC" "#888888"
```

```r
blind_pal <- c("#88CCEE", "#CC6677", "#DDCC77", "#117733", "#332288", "#AA4499",
               "#44AA99", "#999933", "#888888", "#661100", "#6699CC", "#E69F00")

Phylum_RelAbund <- plot_bar(physeq = relative, "Sample", fill = "Phylum") +
  facet_grid(~factor(SampleType, levels = c("Stomach", "Small intestine", "Rectum", "Cloaca","Feces"),
                                           labels= c("Stomach", "Small intestine", "Rectum", "Cloaca",
             scales = "free", space = "free") +
  labs(y="Relative abundance") +
  geom_bar(stat = "identity", position="stack", res=300) +
  scale_fill_manual(values = blind_pal)+ theme(strip.text.x = element_text(face = "bold"),
                                               axis.title.y = element_text(face = "bold")) +
  theme(text = element_text(size = 10))

print(Phylum_RelAbund)
```




```
#ggsave("Samples_DT_Phylum_grammicus.png", width=7.2, height=4.5, dpi=300)
```

Genera

```
metadata <- read.csv(file = "../Data/metadata.csv", header = TRUE, row.names = 1) %>% dplyr::select(Ind
otu_table <- read.csv(file = "../Data/feature_table.csv", check.names = F)
#taxonomy_raw<- read.csv(file = "Genus_Abun_Rel_Sg.csv", check.names = F)
taxonomy <-  read.csv("../Data/taxonomy.csv", check.names = F) %>% mutate_at(
  c("Genus"), str_replace,"g__", "")


otutable_metadata <- otu_table  %>%
  inner_join(taxonomy)
```

```
## Joining with `by = join_by(OTUID)`
```

```
################################################################################
# Calcular la abundancia relativa por género y SampleID
Genus_01 <- otutable_metadata %>% group_by(Genus) %>% summarise_if(is.numeric, sum)
Genus_01 <- Genus_01[c(-1:-2),]


Genus_01 <- Genus_01 %>% column_to_rownames(var = "Genus")
Genus.ra <- t(t(Genus_01)/colSums(Genus_01)*100)
lista <- rowMeans(Genus.ra) %>% as.data.frame() %>% arrange(desc(.)) %>%
  slice_head(n=13) %>% rownames_to_column(var = "Genus") %>%
  filter(!Genus =="g__") %>% filter(!Genus == "Unassigned") %>%
```

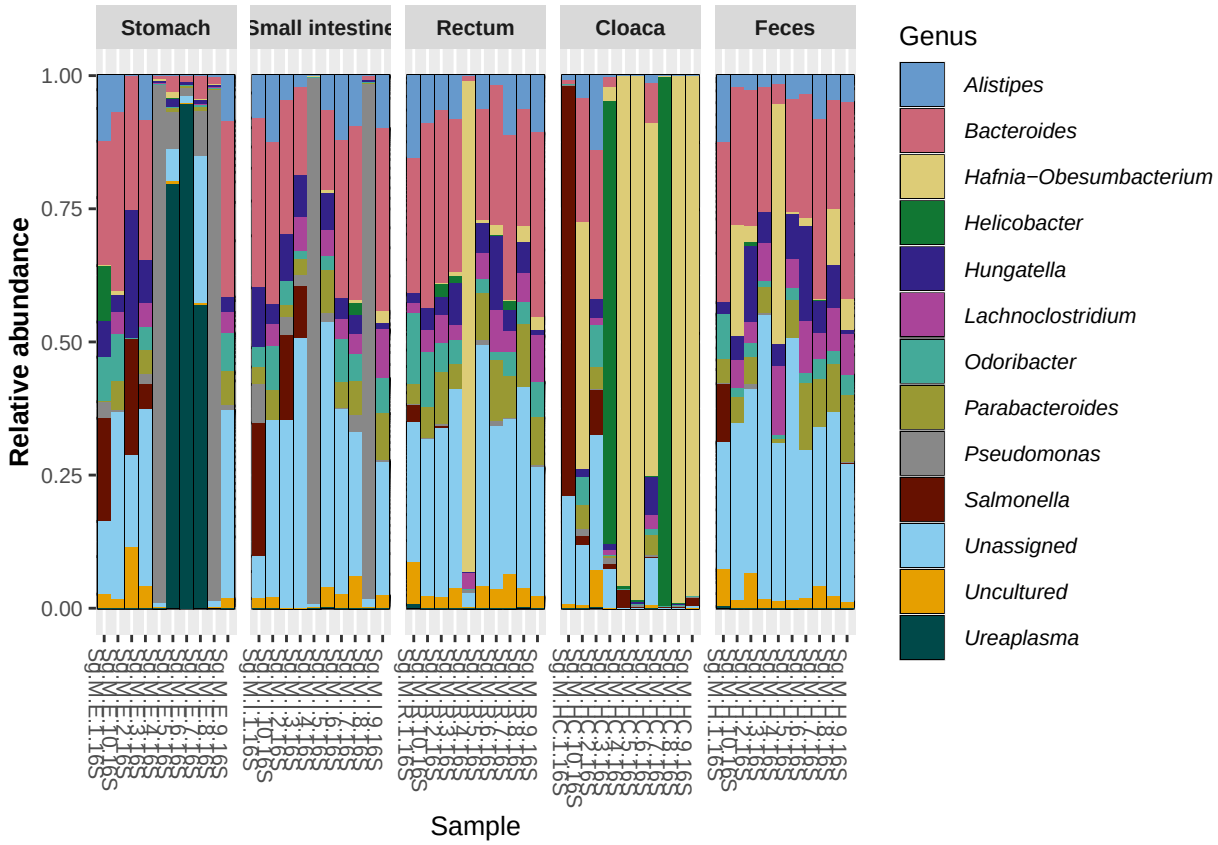
```
  mutate_at(c("Genus"), str_replace,"g__", "")
list <- lista$Genus
#lista01 <- read.csv(file = "lista.csv", check.names = F)
list02 <- lista$Genus
#write.table(lista, file="./lista.txt", sep = "\t")

taxonomy_filter <- taxonomy %>% filter(Genus %in% list02)
taxonomy_1 <- taxonomy_filter %>% inner_join(otu_table, by =c(
  "OTUID"="OTUID")) %>% dplyr::select(1:8)

otu_table_1 <- read.csv(file = "../Data/feature_table.csv", header = TRUE,
                        row.names = 1) %>% rownames_to_column(var = "OTUID") %>%
  inner_join(taxonomy_1, by = "OTUID") %>% dplyr::select(-52:-58) %>%
  column_to_rownames(var = "OTUID")

taxo <- taxonomy_1 %>% column_to_rownames(var = "OTUID")

# Crear objeto de categoría phyloseq
SAM <- sample_data(metadata)
TAX <- tax_table(as.matrix(taxo))
OTU <- otu_table(otu_table_1, taxa_are_rows=TRUE)
#PHY<-phy_tree(phylo)
physeq <- merge_phyloseq(OTU, TAX, SAM)



# Convertir los recuentos de OTUs en abundancias relativas y normalizar
# la abundacia de cada OTU
relative  = transform_sample_counts(physeq = physeq, function(OTU) OTU / sum(OTU))

# Remover las bacterias sin identificacion a nivel Kingdom
physeq_sub <- subset_taxa(physeq, !is.na(Kingdom) & !Kingdom %in% c("", "Unassigned"))
physeq_sub <- subset_taxa(physeq, !is.na(Genus) & !Genus %in% c("", "Unassigned"))

##############################################################################
paleta <- c("#6699CC", "#CC6677", "#DDCC77", "#117733", "#332288", "#AA4499",
            "#44AA99", "#999933", "#888888", "#661100", "#88CCEE", "#E69F00",
            "#004949")


Final_Genus_Sg <- plot_bar(physeq = relative, "Sample", fill = "Genus") +
  facet_grid(~factor(SampleType, levels = c("Stomach", "Small intestine", "Rectum",
                                            "Cloaca", "Feces"),
                     labels = c("Stomach", "Small intestine", "Rectum", "Cloaca",
                                "Feces")), scales = "free", space = "free") +
  labs(y="Relative abundance") +
  geom_bar(stat = "identity", position = "stack", res=300) +
  scale_fill_manual(values = paleta) +
  theme(legend.text = element_text(face = "italic")) +
  scale_fill_manual(values = paleta) +
  theme(strip.text.x = element_text(face = "bold"),
        axis.title.y = element_text(face = "bold")) +
  theme(text = element_text(size = 10))
```

```
## Warning in psmelt(physeq): The sample variables:
## Species
##  have been renamed to:
## sample_Species
## to avoid conflicts with taxonomic rank names.
```

```
## Warning in geom_bar(stat = "identity", position = "stack", res = 300): Ignoring
## unknown parameters: ‘res‘
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
print(Final_Genus_Sg)
```




```
#ggsave("Final_Genus_Sg.jpeg", width=7.2, height=4.8, dpi=300)
```

ALDEx2

```
library(tidyverse)
library(compositions)
library(zCompositions)
#library(CoDaSeq)
library(cowplot)
```

```
library(tidyverse)
library(compositions)
library(zCompositions)
library(CoDaSeq)
```

```
otutable <- read.csv("feature_table.csv", check.names = F, row.names = 1)
metadata <- read.csv("metadata.csv", check.names = F)
taxonomy <- read.csv("taxonomy.csv", check.names = F)

# Creas las tablas pareadas para las muestras letales y no letales
# FECES VS STOMACH
OTUT_FECES <- otutable %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_STOMACH <- otutable %>% dplyr::select_at(vars(contains("M.E.")))
OTUT_fECES_STOMACH <- cbind(OTUT_FECES, OTUT_STOMACH)
write.table(OTUT_fECES_STOMACH, file="./ALDEXGLM_fECES_STOMACH.txt", sep = "\t")

# FECES VS INTESTINE
OTUT_FECES <- otutable %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_INTESTINE <- otutable %>% dplyr::select_at(vars(contains("M.I.")))
OTUT_fECES_INTESTINE <- cbind(OTUT_FECES, OTUT_INTESTINE)
write.table(OTUT_fECES_INTESTINE, file="./ALDEXGLM_fECES_INTESTINE.txt", sep = "\t")

# FECES VS RECTUM
OTUT_FECES <- otutable %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_RECTUM <- otutable %>% dplyr::select_at(vars(contains("M.R.")))
OTUT_fECES_RECTUM <- cbind(OTUT_FECES, OTUT_RECTUM)
write.table(OTUT_fECES_RECTUM, file="./ALDEXGLM_fECES_RECTUM.txt", sep = "\t")

# CLOACA VS STOMACH
OTUT_CLOACA <- otutable %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_STOMACH <- otutable %>% dplyr::select_at(vars(contains("M.E.")))
OTUT_CLOACA_STOMACH <- cbind(OTUT_CLOACA, OTUT_STOMACH)
write.table(OTUT_CLOACA_STOMACH, file="./ALDEXGLM_CLOACA_STOMACH.txt", sep = "\t")

# CLOACA VS INTESTINE
OTUT_CLOACA <- otutable %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_INTESTINE <- otutable %>% dplyr::select_at(vars(contains("M.I.")))
OTUT_CLOACA_INTESTINE <- cbind(OTUT_CLOACA, OTUT_INTESTINE)
write.table(OTUT_CLOACA_INTESTINE, file="./ALDEXGLM_CLOACA_INTESTINE.txt", sep = "\t")

# CLOACA VS RECTUM
OTUT_CLOACA <- otutable %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_RECTUM <- otutable %>% dplyr::select_at(vars(contains("M.R.")))
OTUT_CLOACA_RECTUM <- cbind(OTUT_CLOACA, OTUT_RECTUM)
write.table(OTUT_CLOACA_RECTUM, file="./ALDEXGLM_CLOACA_RECTUM.txt", sep = "\t")

Feces_Stomach <- read.delim("ALDEXGLM_fECES_STOMACH.txt", check.names = F, row.names = 1)
Feces_Intestine <- read.delim("ALDEXGLM_fECES_INTESTINE.txt", check.names = F, row.names = 1)
Feces_Rectum <- read.delim("ALDEXGLM_fECES_RECTUM.txt", check.names = F, row.names = 1)
Cloaca_Stomach <- read.delim("ALDEXGLM_CLOACA_STOMACH.txt", check.names = F, row.names = 1)
Cloaca_Intestine <- read.delim("ALDEXGLM_CLOACA_INTESTINE.txt", check.names = F, row.names = 1)
Cloaca_Rectum <- read.delim("ALDEXGLM_CLOACA_RECTUM.txt", check.names = F, row.names = 1)

library(ALDEx2)
############################
### Feces versus Stomach ###
############################
```

```
covar_FvsS <- metadata %>% filter(SampleType=="Feces"|SampleType=="Stomach") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
  Ind, SampleType) %>% mutate(Type= case_when(
    SampleType=="Feces"~ 0,
    SampleType=="Stomach"~1)) %>% rownames_to_column(var = "id") %>%
  arrange(desc(id)) %>% column_to_rownames(var = "id")
matrix_FvsS <- model.matrix(~SampleType+Ind, data = covar_FvsS)

aldex_clr_FvsS <- aldex.clr(Feces_Stomach, matrix_FvsS, mc.samples = 1000,
                            denom = "all")
aldex_glm_FvsS <- aldex.glm(aldex_clr_FvsS, matrix_FvsS)
aldex_effect_FvsS <- aldex.glm.effect(aldex_clr_FvsS)

aldex_effect_FvsS_type <- as.data.frame(aldex_effect_FvsS) %>%
  rownames_to_column(var = "OTUID")

aldex_table_FvsS <- aldex_glm_FvsS %>% dplyr::select(
  pvalue="model.SampleTypeStomach Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID") %>% inner_join(taxonomy) %>%
  inner_join(aldex_effect_FvsS_type)

write.table(aldex_table_FvsS, file="./GLMaldexFvsS.txt", sep = "\t")

####################################
### Feces versus Small intestine ###
####################################
covar_FvsI <- metadata %>% filter(SampleType=="Feces"|SampleType=="Small intestine") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
  Ind, SampleType) %>% mutate(Type= case_when(
    SampleType=="Feces"~ 0,
    SampleType=="Small intestine"~1)) %>% rownames_to_column(var = "id") %>%
  arrange((id)) %>% column_to_rownames(var = "id")
matrix_FvsI <- model.matrix(~SampleType+Ind, data = covar_FvsI)

aldex_clr_FvsI <- aldex.clr(Feces_Intestine, matrix_FvsI, mc.samples = 1000,
                            denom = "all")
aldex_glm_FvsI <- aldex.glm(aldex_clr_FvsI, matrix_FvsI)
aldex_effect_FvsI <- aldex.glm.effect(aldex_clr_FvsI)

aldex_effect_FvsI_type <- as.data.frame(aldex_effect_FvsI) %>%
  rownames_to_column(var = "OTUID")
aldex_table_FvsI <-  aldex_glm_FvsI %>% dplyr::select(
  pvalue="model.SampleTypeSmall intestine Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID")  %>%
  inner_join(taxonomy) %>% inner_join(aldex_effect_FvsI_type)

write.table(aldex_table_FvsI, file="./GLMaldexFvsI.txt", sep = "\t")

############################
### Feces versus Rectum ###
############################
covar_FvsR <- metadata %>% filter(SampleType=="Feces"|SampleType=="Rectum") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
```

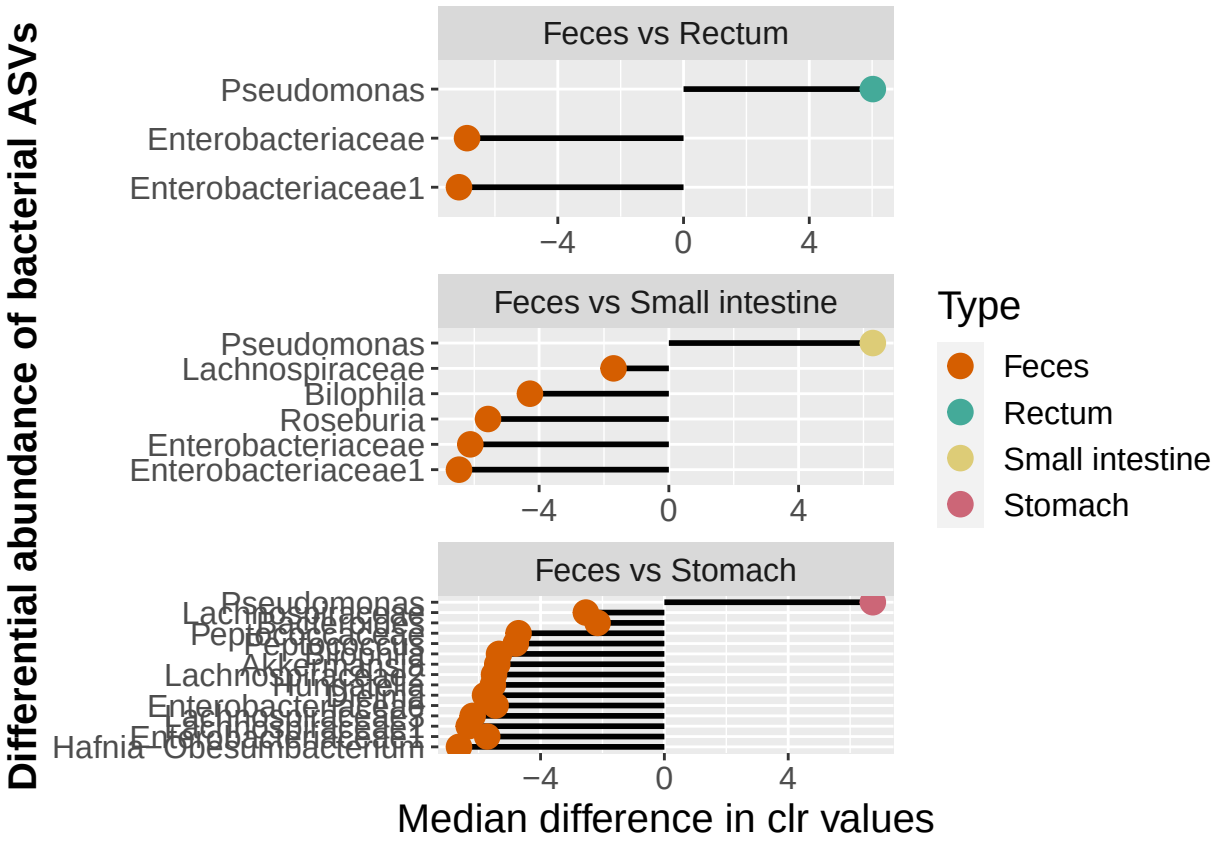
```
  Ind, SampleType) %>% mutate(Type= case_when(
    SampleType=="Feces"~ 0,
    SampleType=="Rectum"~1)) %>% rownames_to_column(var = "id") %>%
  arrange((id)) %>% column_to_rownames(var = "id")
matrix_FvsR <- model.matrix(~SampleType+Ind, data = covar_FvsR)

aldex_clr_FvsR <- aldex.clr(Feces_Rectum, matrix_FvsR, mc.samples = 1000,
                            denom = "all")
aldex_glm_FvsR <- aldex.glm(aldex_clr_FvsR, matrix_FvsR)
aldex_effect_FvsR <- aldex.glm.effect(aldex_clr_FvsR)

aldex_effect_FvsR_type <- as.data.frame(aldex_effect_FvsR) %>%
  rownames_to_column(var = "OTUID")
aldex_table_FvsR <-  aldex_glm_FvsR %>% dplyr::select(
  pvalue="model.SampleTypeRectum Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID")  %>% inner_join(taxonomy) %>%
  inner_join(aldex_effect_FvsR_type)

write.table(aldex_table_FvsR, file="./GLMaldexFvsR.txt", sep = "\t")

##############################
### Cloaca versus Stomach ###
##############################
covar_CvsS <- metadata %>% filter(SampleType=="Cloaca"|SampleType=="Stomach") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
  Ind, SampleType) %>% mutate(SampleType= case_when(
    SampleType=="Swab"~ "Cloaca",
    TRUE ~ as.character(SampleType))) %>% mutate(Type= case_when(
      SampleType=="Cloaca"~ 0,
      SampleType=="Stomach"~1)) %>% rownames_to_column(var = "id") %>%
  arrange(desc(id)) %>% column_to_rownames(var = "id")
matrix_CvsS <- model.matrix(~SampleType+Ind, data = covar_CvsS)

aldex_clr_CvsS <- aldex.clr(Cloaca_Stomach, matrix_CvsS, mc.samples = 1000,
                            denom = "all")
aldex_glm_CvsS <- aldex.glm(aldex_clr_CvsS, matrix_CvsS)
aldex_effect_CvsS <- aldex.glm.effect(aldex_clr_CvsS)

aldex_effect_CvsS_type <- as.data.frame(aldex_effect_CvsS) %>%
  rownames_to_column(var = "OTUID")
aldex_table_CvsS <-  aldex_glm_CvsS %>% dplyr::select(
  pvalue="model.SampleTypeStomach Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID")  %>%
  inner_join(taxonomy) %>% inner_join(aldex_effect_CvsS_type)

write.table(aldex_table_CvsS, file="./GLMaldexCvsS.txt", sep = "\t")

#####################################
### Cloaca versus Small intestine ###
#####################################
covar_CvsI <- metadata %>%
  filter(SampleType=="Cloaca"|SampleType=="Small intestine") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
```

```
  Ind, SampleType) %>% mutate(SampleType= case_when(
    SampleType=="Swab"~ "Cloaca",
    TRUE ~ as.character(SampleType)))%>% mutate(Type= case_when(
    SampleType=="Cloaca"~ 0,
    SampleType=="Small intestine"~1)) %>% rownames_to_column(var = "id") %>%
  arrange((id)) %>% column_to_rownames(var = "id")
matrix_CvsI <- model.matrix(~SampleType+Ind, data = covar_CvsI)

aldex_clr_CvsI <- aldex.clr(Cloaca_Intestine, matrix_CvsI, mc.samples = 1000,
                            denom = "all")
aldex_glm_CvsI <- aldex.glm(aldex_clr_CvsI, matrix_CvsI)
aldex_effect_CvsI <- aldex.glm.effect(aldex_clr_CvsI)

aldex_effect_CvsI_type <- as.data.frame(aldex_effect_CvsI) %>%
  rownames_to_column(var = "OTUID")
aldex_table_CvsI <-  aldex_glm_CvsI %>% dplyr::select(
  pvalue="model.SampleTypeSmall intestine Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID")  %>%
  inner_join(taxonomy) %>% inner_join(aldex_effect_CvsI_type)

write.table(aldex_table_CvsI, file="./GLMaldexCvsI.txt", sep = "\t")

############################
### Cloaca versus Rectum ###
############################
covar_CvsR <- metadata %>% filter(SampleType=="Cloaca"|SampleType=="Rectum") %>%
  column_to_rownames(var = "SampleID") %>% dplyr::select(
  Ind, SampleType) %>% mutate(SampleType= case_when(
    SampleType=="Swab"~ "Cloaca",
    TRUE ~ as.character(SampleType)))%>% mutate(Type= case_when(
      SampleType=="Cloaca"~ 0,
      SampleType=="Rectum"~1)) %>% rownames_to_column(var = "id") %>%
  arrange((id)) %>% column_to_rownames(var = "id")
matrix_CvsR <- model.matrix(~SampleType+Ind, data = covar_CvsR)

aldex_clr_CvsR <- aldex.clr(Cloaca_Rectum, matrix_CvsR, mc.samples = 1000,
                            denom = "all")
aldex_glm_CvsR <- aldex.glm(aldex_clr_CvsR, matrix_CvsR)
aldex_effect_CvsR <- aldex.glm.effect(aldex_clr_CvsR)

aldex_effect_CvsR_type <- as.data.frame(aldex_effect_CvsR) %>%
  rownames_to_column(var = "OTUID")
aldex_table_CvsR <-  aldex_glm_CvsR %>% dplyr::select(
  pvalue="model.SampleTypeRectum Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "OTUID") %>%
  inner_join(taxonomy)%>% inner_join(aldex_effect_CvsR_type)

write.table(aldex_table_CvsR, file="./GLMaldexCvsR.txt", sep = "\t")
```

Aldex Plot

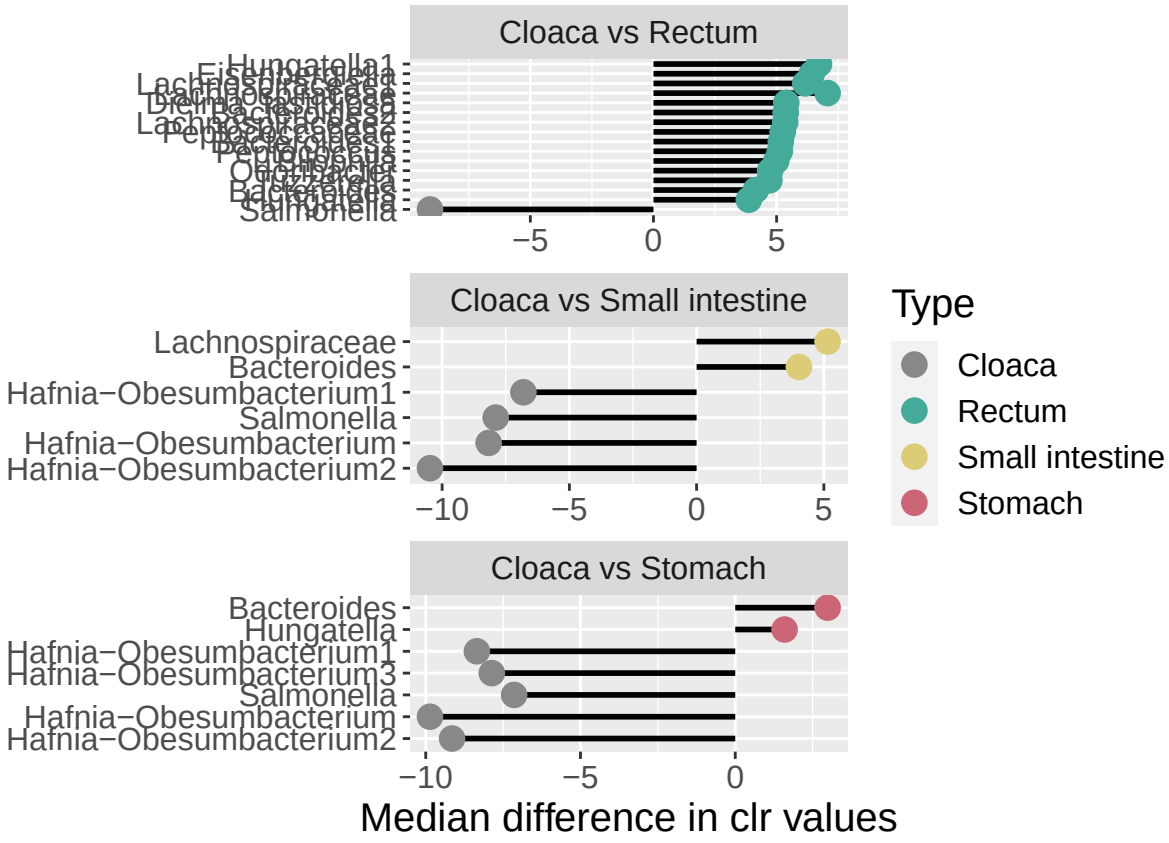
```
# Graficar Feces vs DT segments
GLMaldexFvsS <- read.delim("../Data/GLMaldexFvsS.txt", check.names = F)
```

```
GLMaldexFvsI <- read.delim("../Data/GLMaldexFvsI.txt", check.names = F)
GLMaldexFvsR <- read.delim("../Data/GLMaldexFvsR.txt", check.names = F)

p1 <- GLMaldexFvsS %>% mutate(Type = case_when(
    diff.btw >0  ~"Stomach", diff.btw <0 ~"Feces")) %>%
  mutate(Compare="Feces vs Stomach") %>% rename(Other="Stomach")

p2 <- GLMaldexFvsI %>% mutate(Type = case_when(
  diff.btw >0  ~"Small intestine", diff.btw <0 ~"Feces")) %>%
  mutate(Compare="Feces vs Small intestine") %>% rename(Other="Small intestine")

p3 <- GLMaldexFvsR %>% mutate(Type = case_when(
  diff.btw >0  ~"Rectum", diff.btw <0 ~"Feces")) %>%
  mutate(Compare="Feces vs Rectum") %>% rename(Other="Rectum")

pn <- rbind(p1, p2, p3)

plot1 <- pn %>% arrange(diff.btw) %>% #filter(!effect>abs(1)) %>%
  ggplot(., aes(x=diff.btw, y=reorder(taxonomy, diff.btw), fill=Type)) +
  geom_bar(stat = "identity", width = 0.5) +
  facet_wrap(~Compare, ncol = 1, scales = "free") +
  theme(text = element_text(size = 15)) +
  ylab("Differential abundance of bacterial ASVs") +
  scale_y_discrete(expand = c(0,0))

plot2_1 <- pn %>% arrange(diff.btw) %>% #filter(!effect>abs(1)) %>%
  ggplot(., aes(x=diff.btw, y=reorder(taxonomy, diff.btw), fill=Type)) +
  geom_segment(aes(yend=reorder(taxonomy, diff.btw), xend=0), size=1) +
  geom_point(size=4, aes(colour=Type)) +
  facet_wrap(~Compare, ncol = 1, scales = "free") +
  theme(text = element_text(size = 15),
        axis.title.y = element_text(face = "bold"),
        legend.position = "right") +
  ylab("Differential abundance of bacterial ASVs") + scale_color_manual(
    values = c("#D55E00","#44AA99", "#DDCC77", "#CC6677")) +
  xlab("Median difference in clr values")
print(plot2_1)
```




```
# Graficar Cloaca vs DT segments
GLMaldexCvsS <- read.delim("../Data/GLMaldexCvsS.txt", check.names = F)
GLMaldexCvsI <- read.delim("../Data/GLMaldexCvsI.txt", check.names = F)
GLMaldexCvsR <- read.delim("../Data/GLMaldexCvsR.txt", check.names = F)

C1 <- GLMaldexCvsS %>% mutate(Type = case_when(
  diff.btw >0 ~"Stomach", diff.btw <0 ~"Cloaca")) %>%
  mutate(Compare="Cloaca vs Stomach") %>% dplyr::select(everything(),Other=Stomach)

C2 <- GLMaldexCvsI %>% mutate(Type = case_when(
  diff.btw >0 ~"Small intestine", diff.btw <0 ~"Cloaca")) %>%
  mutate(Compare="Cloaca vs Small intestine") %>% dplyr::rename(
    Other="Small intestine")

C3 <- GLMaldexCvsR %>% mutate(Type = case_when(
  diff.btw >0  ~"Rectum", diff.btw <0 ~"Cloaca")) %>%
  mutate(Compare="Cloaca vs Rectum") %>% dplyr::rename(Other="Rectum")

CN <- rbind(C1,C2,C3)

plot2 <- CN %>% arrange(diff.btw) %>% #filter(!effect>abs(1)) %>%
  ggplot(., aes(x=diff.btw, y=reorder(taxonomy, diff.btw), fill=Type)) +
  geom_bar(stat = "identity", width = 0.8) +
  facet_wrap(~Compare, ncol = 1, scales = "free") +
  theme(text = element_text(size = 15))

plot2_2 <- CN %>% arrange(diff.btw) %>% #filter(!effect>abs(1)) %>%
```

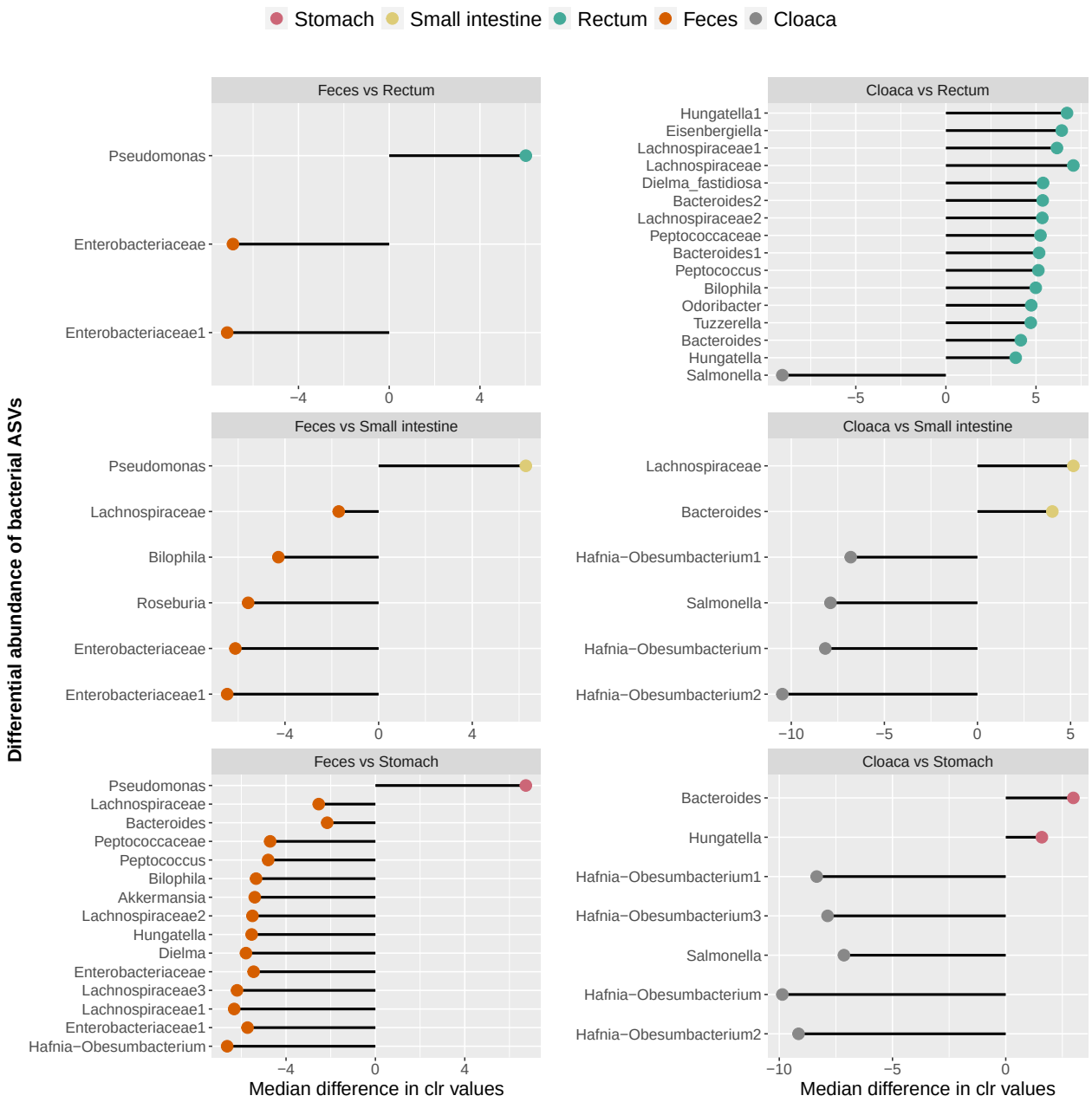
```
  ggplot(., aes(x=diff.btw, y=reorder(taxonomy, diff.btw), fill=Type)) +
  geom_segment(aes(yend=reorder(taxonomy, diff.btw), xend=0), size=1) +
  geom_point(size=4, aes(colour=Type)) +
  facet_wrap(~Compare, ncol = 1, scales = "free") +
  theme(text = element_text(size = 15)) +
  ylab("") + scale_color_manual(
    values = c("#888888","#44AA99", "#DDCC77", "#CC6677")) +
  xlab("Median difference in clr values")
print(plot2_2)
```




```
# Create legend
library(ggpubr)
alpha <- read.csv("../Data/Hill_numbers_q012.csv") %>% dplyr::select(SampleID, q0, q1, q2)
metadata <- read.csv("../Data/metadata.csv",check.names = F) %>% dplyr::select(SampleID:Ta)
alpha <- alpha %>% inner_join(metadata, by = c("SampleID"="SampleID"))

leg_order <- c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca")
leg <- alpha %>% ggplot(aes(x = factor(SampleType, level=leg_order), y = q1,
                            color=factor(SampleType, level=leg_order))) +
  geom_point(size=4) +
  scale_color_manual(values = c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888")) +
  theme(legend.position = "top", legend.direction = "horizontal",
        legend.title = element_blank(), legend.text = element_text(size = 16))
legends <- get_legend(leg)
```

```
#plot2
library(cowplot)
```

```
leg2 <- plot_grid(NULL, legends, NULL, ncol = 3)
b <- plot_grid(plot2_1 + theme(legend.position = "none"),
               plot2_2 + theme(legend.position = "none"),
               rel_widths = c(1,1))

plot_aldex <- plot_grid(plot2_1 + theme(legend.position = "none"),
               plot2_2 + theme(legend.position = "none"),
             rel_widths = c(1,1))

plot_aldex <- plot_grid(leg2, b, nrow = 2, rel_heights = c(0.1,1))
plot_aldex
```




```
#ggsave(plot = plot_aldex, "Plot_ALDEX2glm.jpg", width = 12, height = 12.5)
```

```
#fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
```

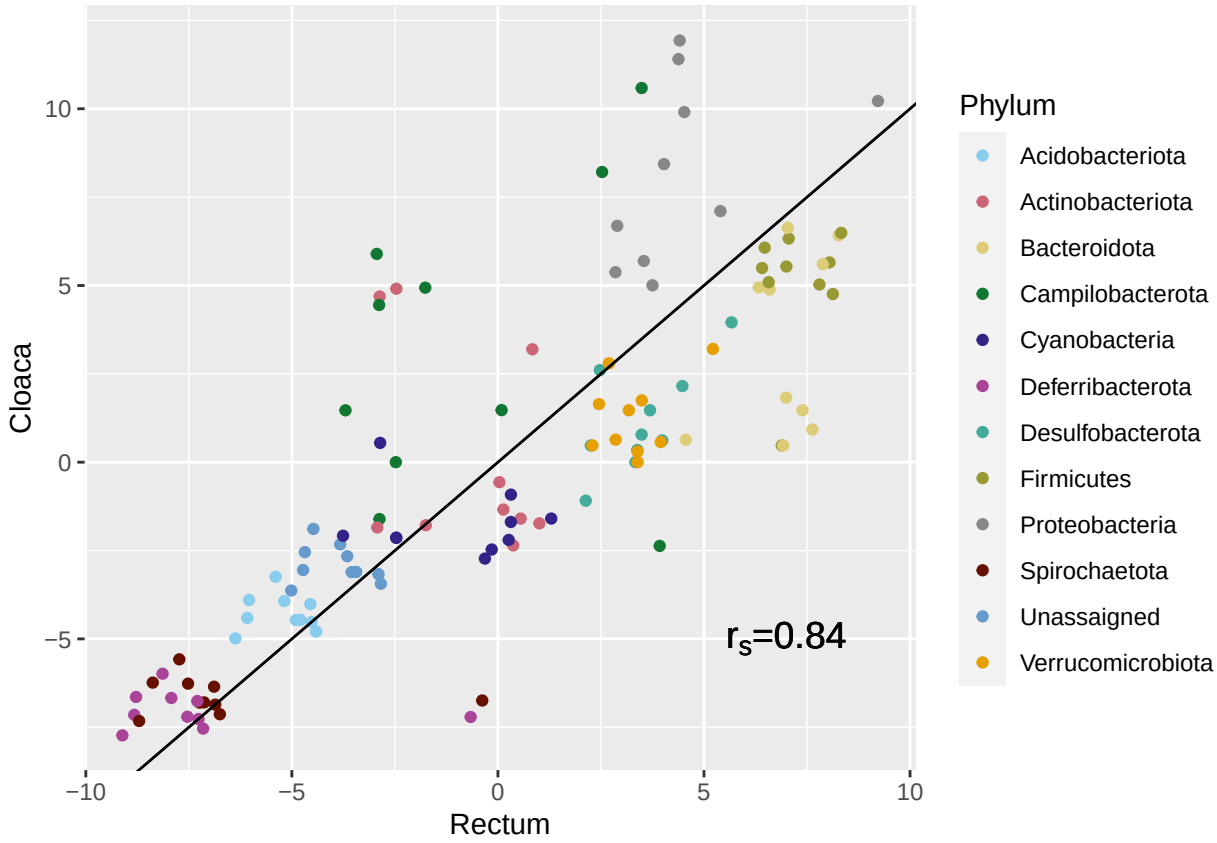
Linear Regression

```
# Linear regression
# Laod packages
library(tidyverse)
library(CoDaSeq)
library(zCompositions)
library(compositions)
library(propr)
library(CoDaSeq)

#Load files
phyl <- read_csv("../Data/level-2.csv")
phyl2 <- phyl %>% dplyr::select(index, contains("d__")) %>%
  column_to_rownames(var = "index")
# Create pallete color
blind_pal <- c("#88CCEE", "#CC6677", "#DDCC77", "#117733", "#332288", "#AA4499",
               "#44AA99", "#999933", "#888888", "#661100", "#6699CC", "#E69F00")
```

```
d.pro <- cmultRepl(t(phyl2), method = "CZM", output = "p-counts")
d.clr.abund.codaseq <- codaSeq.clr(x= d.pro, samples.by.row = F)
############################
### Cloaca versus Rectum ###
############################
phyl_S_R <- data.frame(t(d.clr.abund.codaseq)) %>%
  rownames_to_column(var = "index") %>% inner_join(phyl) %>%
  dplyr::select(c(1:13), SampleType) %>% filter(
      SampleType=="Rectum"|SampleType=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_S_R <- phyl_S_R %>% dplyr::select(-index)
namesotu <- otu_S_R$names
#write_tsv(phyl_S_R, "phyl_S_R.tsv")
```

```
#Load file
SR <- read.csv("../Data/Swab_Rectum.csv")
Cloaca_Rectum <- SR %>% ggplot(aes(x=Rectum, y=Cloaca, color=Phylum)) +
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(SR$Cloaca, SR$Rectum, method = "spearman"),digits = 2)))))
#  labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
 #                   "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
  #                   " Slope =",signif(data.lm_SR$coef[[2]], 5),
   #                   " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
```

```
Cloaca_Rectum
```

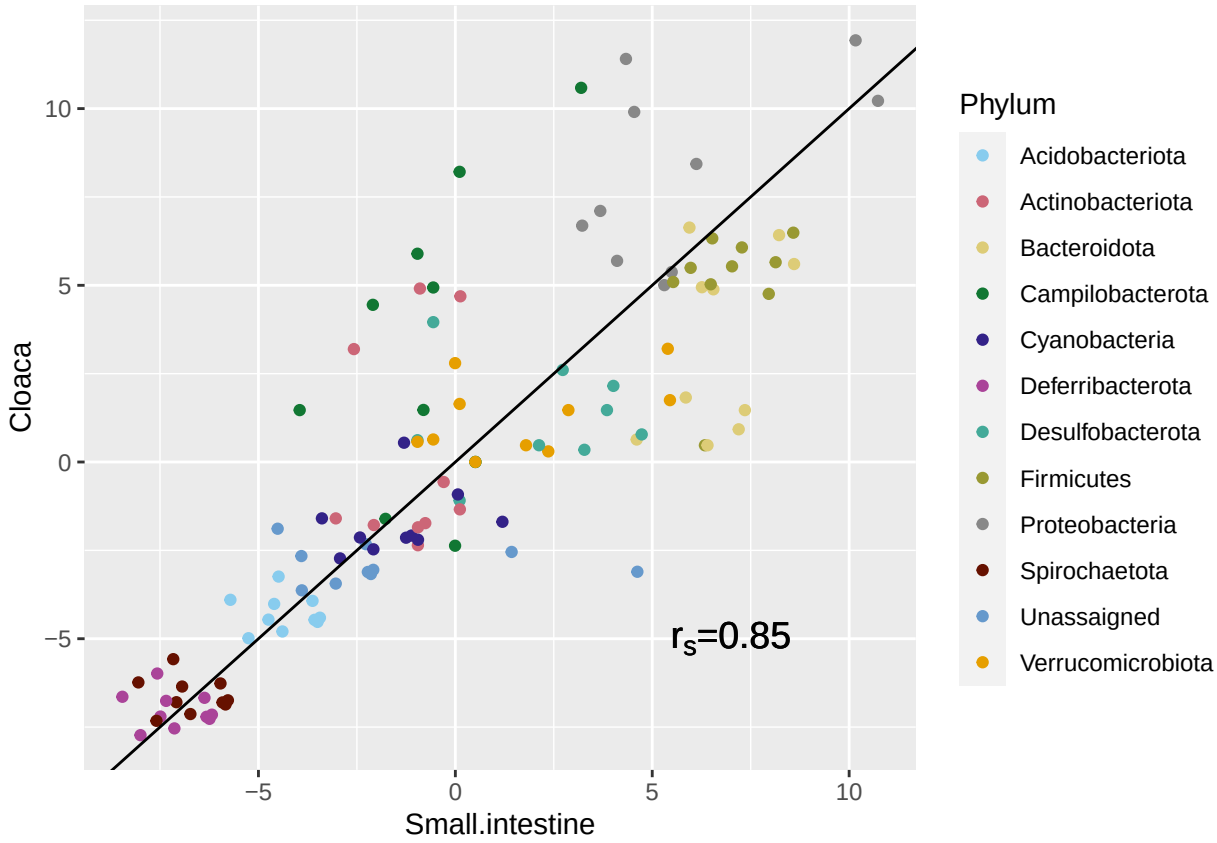



```
#ggsave("Cloaca_Rectum.jpeg", width=7, height=4.5, dpi=300)
```

```
#####################################
### Cloaca versus Small intestine ###
#####################################
phyl_S_SI <- data.frame(t(d.clr.abund.codaseq)) %>%
  rownames_to_column(var = "index") %>%
  inner_join(phyl) %>% dplyr::select(c(1:13), SampleType) %>% filter(
      SampleType=="Smallintestine"|SampleType=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_S_SI <- phyl_S_SI %>% dplyr::select(-index)
namesotu <- otu_S_SI$names
write_tsv(phyl_S_SI, "phyl_S_SI.tsv")
```

```
#Load file
SI <- read.csv("../Data/Swab_Intestine.csv")
Cloaca_Intest <- SI %>% ggplot(aes(x=Small.intestine, y=Cloaca, color=Phylum)) +
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
```

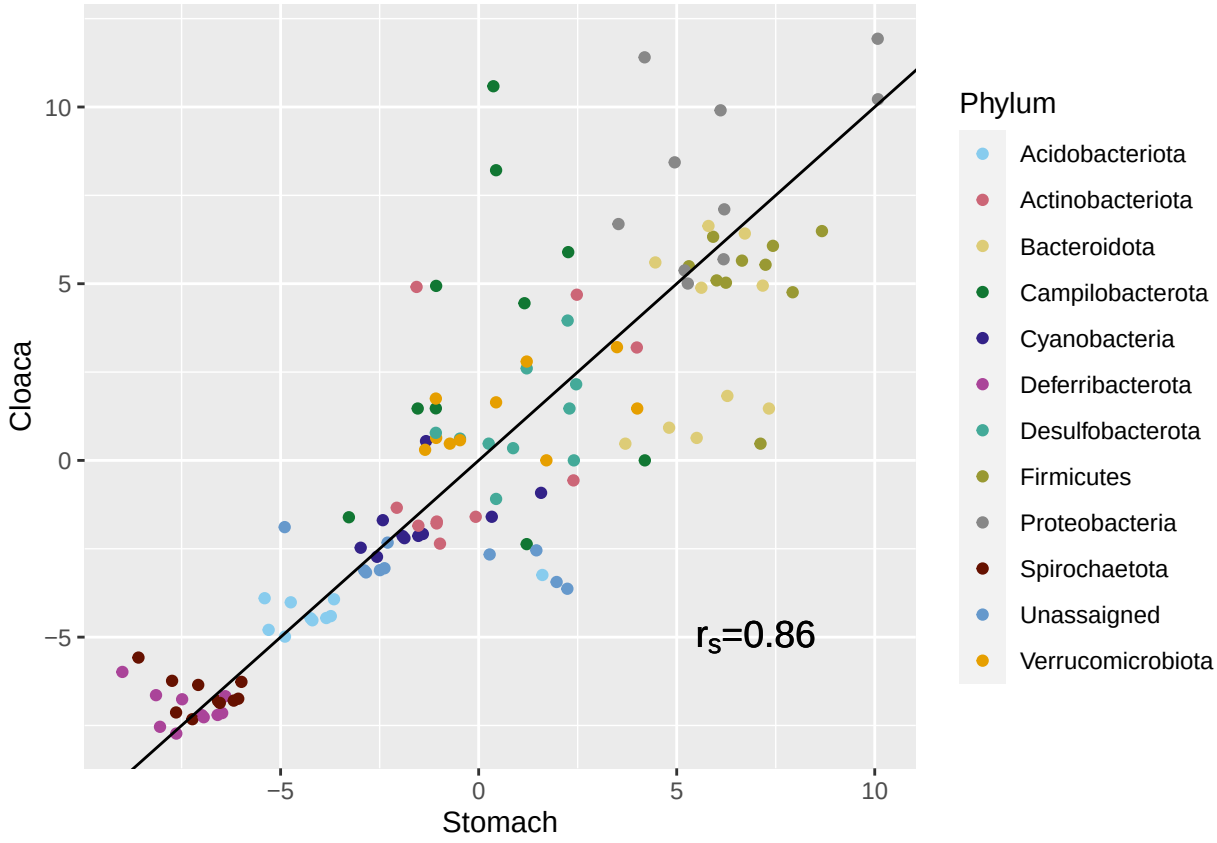
```
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(SI$Cloaca, SI$Small.intestine, method = "spearman"),digits = 2)))))
# labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
#                    "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
#                   " Slope =",signif(data.lm_SR$coef[[2]], 5),
#                  " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
Cloaca_Intest
```




```
##############################
### Cloaca versus Stomach ###
##############################
phyl_S_Sto <- data.frame(t(d.clr.abund.codaseq)) %>%
  rownames_to_column(var = "index") %>%
  inner_join(phyl) %>% dplyr::select(c(1:13), SampleType) %>% filter(
    SampleType=="Stomach"|SampleType=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_S_Sto <- phyl_S_Sto %>% dplyr::select(-index)
namesotu <- otu_S_Sto$names
write_tsv(phyl_S_Sto, "phyl_S_Sto.tsv")
```

```
#Load file
SStom <- read.csv("../Data/Swab_Stomach.csv")
```

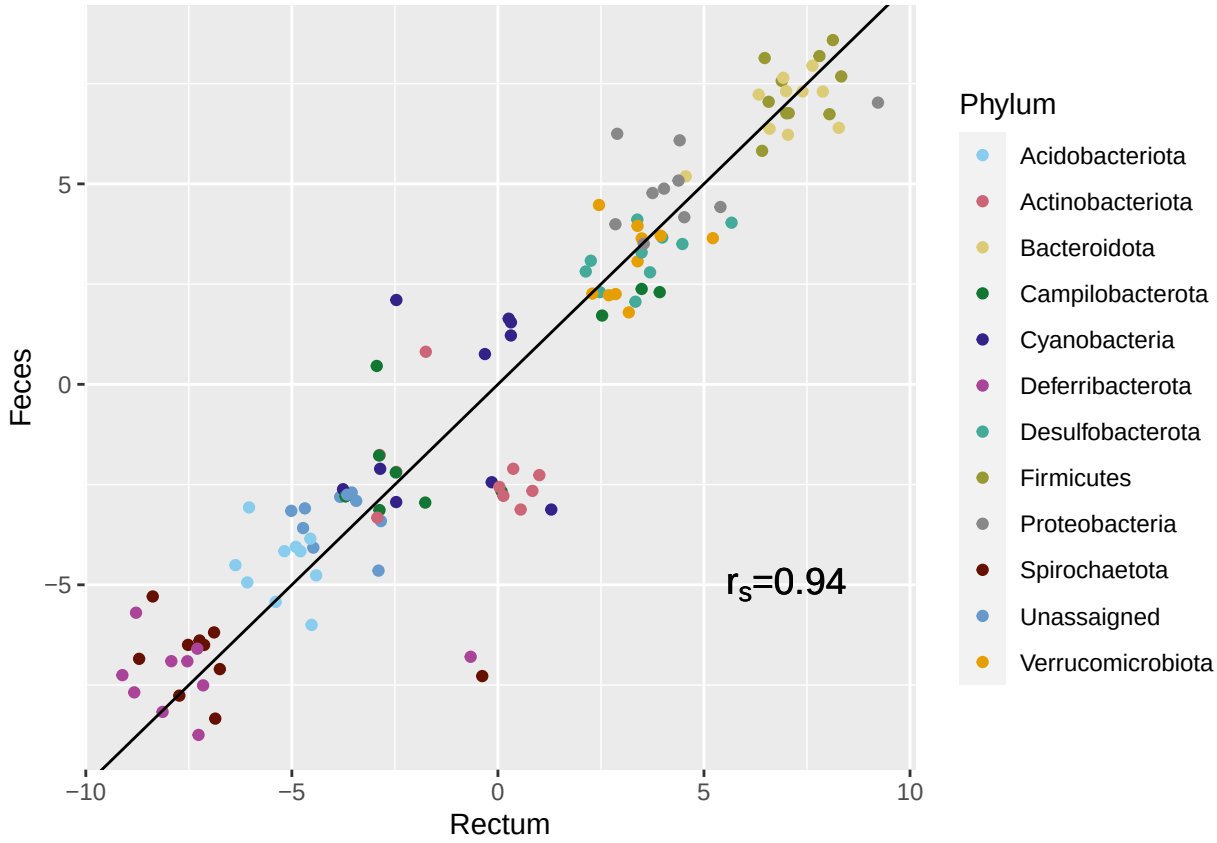
```
Cloaca_Stomach <- SStom %>% ggplot(aes(x=Stomach, y=Cloaca, color=Phylum)) +
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(SStom$Cloaca, SStom$Stomach, method = "spearman"),digits = 2)))))
# labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
#                    "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
#                   " Slope =",signif(data.lm_SR$coef[[2]], 5),
#                  " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
Cloaca_Stomach
```




```
############################
### Feces versus Rectum ###
############################
phyl_F_R <- data.frame(t(d.clr.abund.codaseq)) %>%
  rownames_to_column(var = "index") %>%
  inner_join(phyl) %>% dplyr::select(c(1:13), SampleType) %>% filter(
    SampleType=="Rectum"|SampleType=="Feces") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_F_R <- phyl_F_R %>% dplyr::select(-index)
```

```
namesotu <- otu_F_R$names
write_tsv(phyl_F_R, "phyl_F_R.tsv")
```

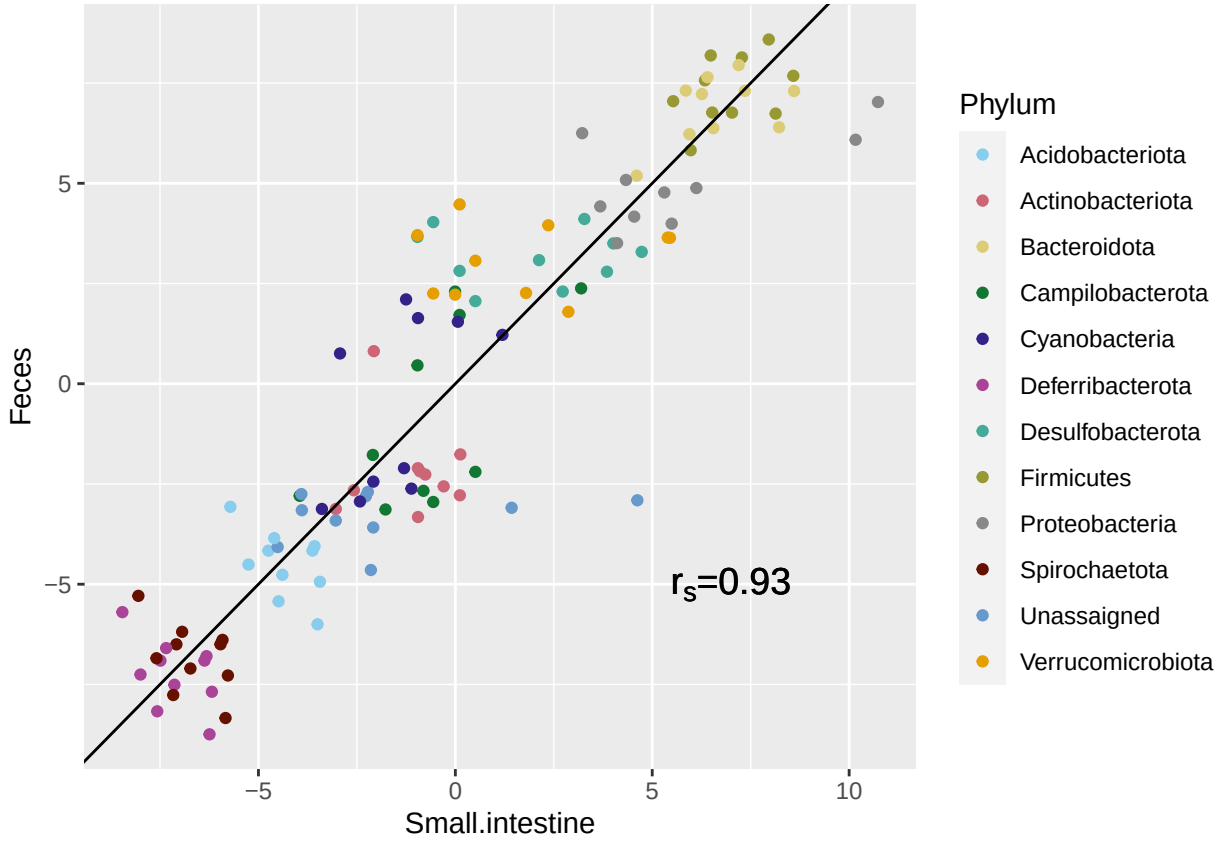
```
# Load file
FR <- read.csv("../Data/Feces_Rectum.csv")
Feces_Rectum <- FR %>% ggplot(aes(x=Rectum, y=Feces, color=Phylum)) +
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(FR$Feces, FR$Rectum, method = "spearman"),digits = 2)))))
# labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
#                    "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
#                   " Slope =",signif(data.lm_SR$coef[[2]], 5),
#                  " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
Feces_Rectum
```




```
####################################
### Feces versus Small intestine ###
####################################
phyl_F_SI <- data.frame(t(d.clr.abund.codaseq)) %>%
  rownames_to_column(var = "index") %>%
  inner_join(phyl) %>% dplyr::select(c(1:13), SampleType) %>% filter(
    SampleType=="Smallintestine"|SampleType=="Feces") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
```

```
                names_to = "names", values_to = "values") %>% pivot_wider(
                  names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_F_SI <- phyl_F_SI %>% dplyr::select(-index)
namesotu <- otu_F_SI$names
write_tsv(phyl_F_SI, "phyl_F_SI.tsv")
```

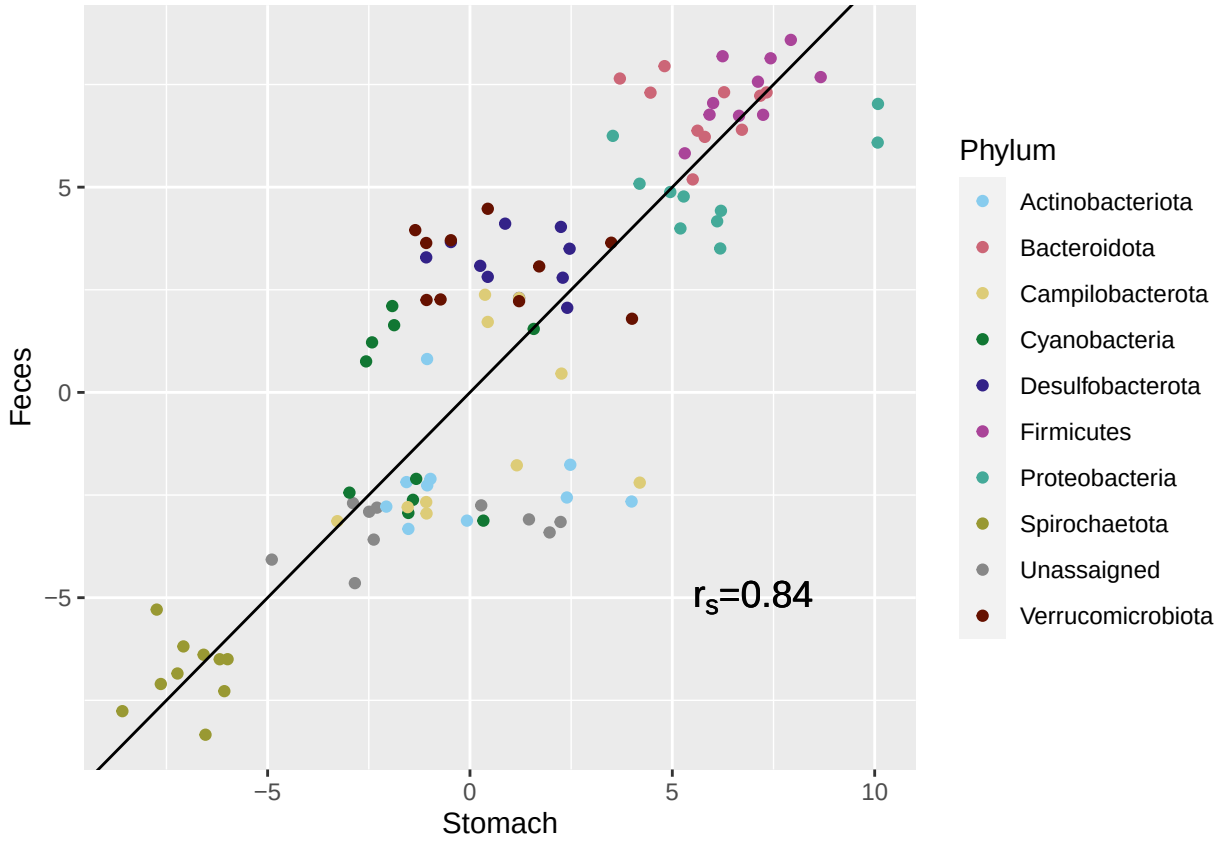
```
# Load file
FI <- read.csv("../Data/Feces_Intestine.csv")
Feces_Intest <- FI %>% ggplot(aes(x=Small.intestine, y=Feces, color=Phylum))+
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(FI$Feces, FI$Small.intestine, method = "spearman"),digits = 2)))))
# labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
#                    "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
#                    " Slope =",signif(data.lm_SR$coef[[2]], 5),
#                    " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
Feces_Intest
```




```
############################
### Feces versus Stomach ###
############################
phyl_F_Sto <- data.frame(t(d.clr.abund.codaseq)) %>%
```

```
  rownames_to_column(var = "index") %>%
  inner_join(phyl) %>% dplyr::select(c(1:11), SampleType) %>% filter(
    SampleType=="Stomach"|SampleType=="Feces") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("d__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>%
  replace(is.na(.), 0)

otu_F_Sto <- phyl_F_Sto %>% dplyr::select(-index)
namesotu <- otu_F_Sto$names
write_tsv(phyl_F_Sto, "phyl_F_Sto.tsv")
```

```
# Load file
FStom <- read.csv("../Data/Feces_Stomach.csv")
Feces_Stomach <- FStom %>% ggplot(aes(x=Stomach, y=Feces, color=Phylum)) +
  geom_point() +
  scale_color_manual(values = blind_pal) +
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", x=7, y=-5, size=5,label=bquote(paste('r'['s']*'=',.(round(
    cor(FStom$Feces, FStom$Stomach, method = "spearman"),digits = 2)))))
# labs(title = paste("Adj R2 = ",signif(summary(data.lm_SR)$adj.r.squared, 5),
#                    "Intercept =",signif(data.lm_SR$coef[[1]],5 ),
#                    " Slope =",signif(data.lm_SR$coef[[2]], 5),
#                    " P =",signif(summary(data.lm_SR)$coef[2,4], 5)))
Feces_Stomach
```

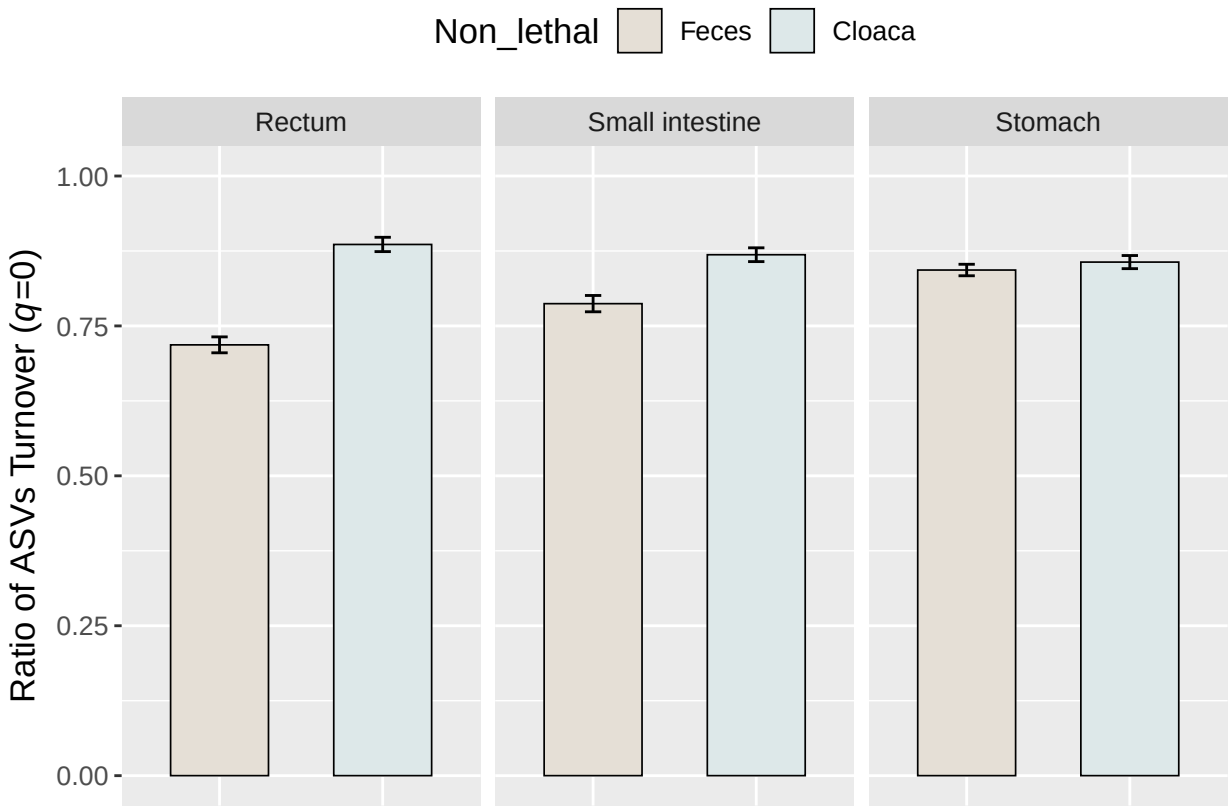


```
#ggsave("Feces_Stomach.jpeg", width=7, height=4.5, dpi=300)
```

TurnOver

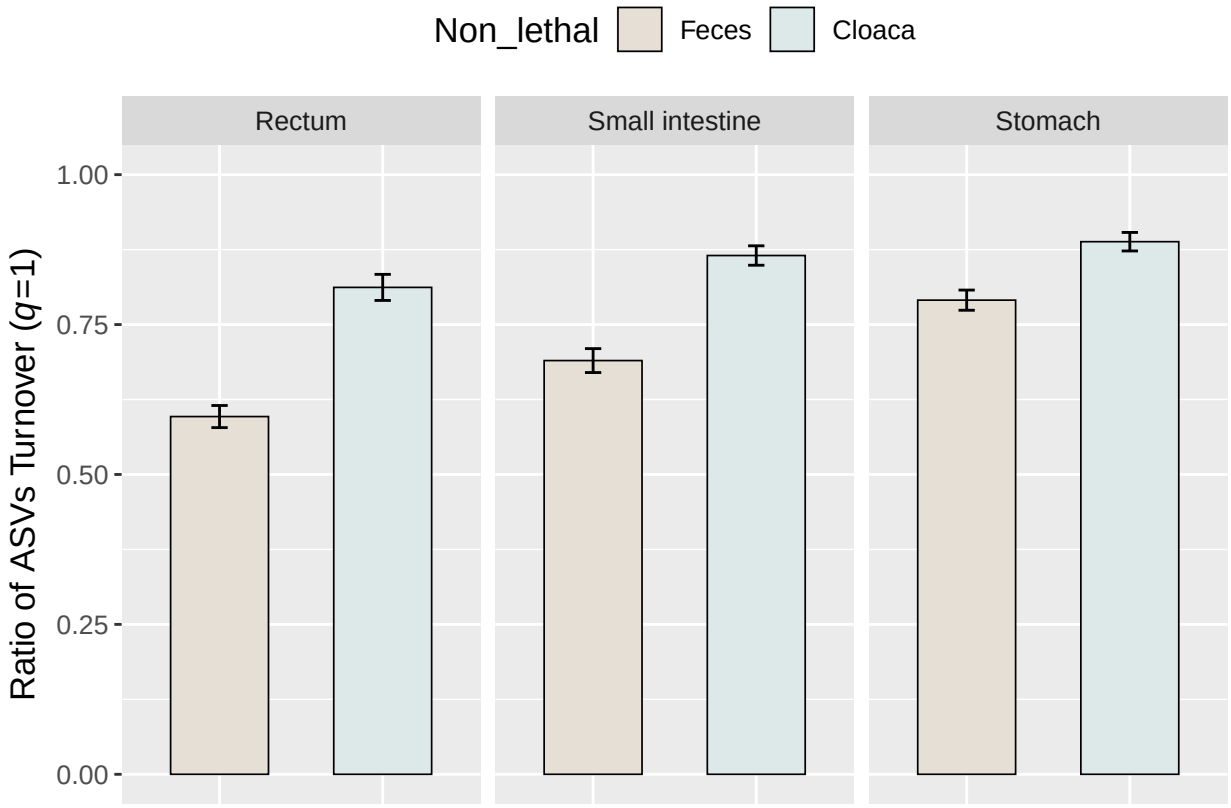
```
library(hillR)
library(dplyr)
library(tidyverse)
library(vegan)
library(ggplot2)
library(ggpubr)
library(hilldiv)

# Load files
otutable <- read.csv("../Data/feature_table.csv", check.names = F, row.names = 1)
metadata <- read.csv("../Data/metadata.csv", check.names = F) %>% dplyr::select(SampleID:Ta)
#q0_hillr <- hill_taxa_parti_pairwise(t(otutable), q = 0) %>%
#  mutate(recambio = TD_beta-1)
#q1 <- hill_taxa_parti_pairwise(t(otutable), q = 1)%>%
#  mutate(recambio = TD_beta-1)
#q2 <- hill_taxa_parti_pairwise(t(otutable), q = 2)%>%
#  mutate(recambio = TD_beta-1)

#q0m <- hill_taxa_parti_pairwise(t(otutable), q = 0, output = "matrix", pairs = "full")
#q1m <- hill_taxa_parti_pairwise(t(otutable), q = 1, output = "matrix", pairs = "full")
#q2m <- hill_taxa_parti_pairwise(t(otutable), q = 2, output = "matrix", pairs = "full")

###################################
## Calculate turnover ratio (q0) ##
###################################
q0 <- pair_dis(otutable, qvalue = 0, metric = "V")
q0_df <- reshape2::melt(q0$L1_VqN, varnames=c("Site1", "Site2"),
                        value.name = "turnover", na.rm=TRUE)

q0_df_meta <- q0_df %>% inner_join(metadata, by = c("Site1"="SampleID")) %>%
  inner_join(metadata, by = c("Site2" = "SampleID")) %>% mutate(
    Types = paste0(SampleType.x, "vs", SampleType.y))
#write.table(q0_df_meta, file="./TURNOVER_q0.txt", sep = "\t")

# RATIO OF ASVs TURNOVER AT q=0
beta <- read.csv("../Data/Inter_q0.csv", header = TRUE, check.names = F)
titulo0 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=0)"))

Turnover_q0 <- ggbarplot(beta, x= "Non_lethal", y= "Turnover",
                         color = "black",  width = 0.6, lwd=0.3,
                         facet.by = "DT", fill = "Non_lethal",
                         add = "mean_se") +
  labs(x= element_blank(), y = titulo0) +
  theme_gray() + theme(text = element_text (size = 13)) +
  theme(legend.position = "right",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  scale_y_continuous(limits = c(0,1)) +
```

```
  theme(legend.position = "top") +
  scale_fill_manual(values=c("#E5DFD6","#DDE8E9"))

print(Turnover_q0)
```



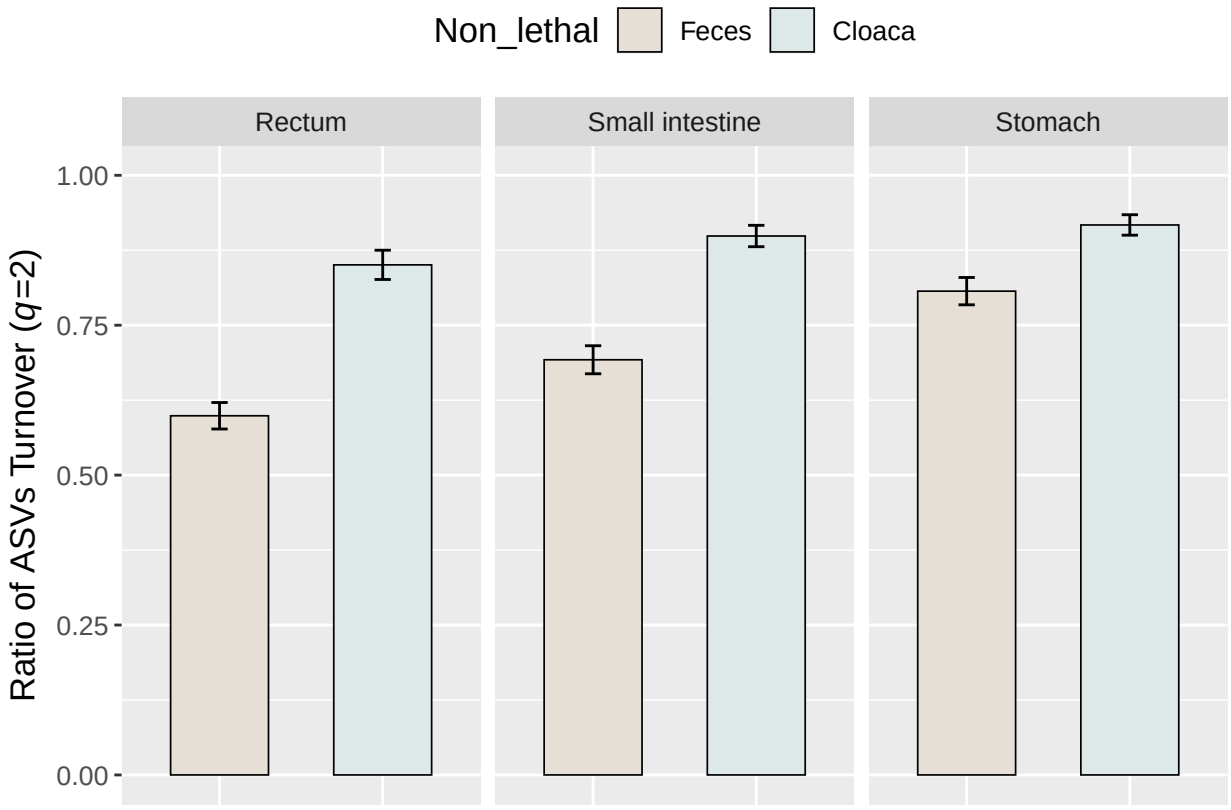

```
###################################
## Calculate turnover ratio (q1) ##
###################################
q1 <- pair_dis(otutable, qvalue = 1, metric = "V")
q1_df <- reshape2::melt(q1$L1_VqN, varnames=c("Site1", "Site2"),
                        value.name = "turnover", na.rm=TRUE)

q1_df_meta <- q1_df %>% inner_join(metadata, by = c("Site1"="SampleID")) %>%
  inner_join(metadata, by = c("Site2" = "SampleID")) %>% mutate(
    Types = paste0(SampleType.x, "vs", SampleType.y))
#write.table(q1_df_meta, file="./TURNOVER_q1.txt", sep = "\t")

# RATIO OF ASVs TURNOVER AT q=1
beta_q1 <- read.csv("../Data/Inter_q1.csv", header = TRUE, check.names = F)
titulo1 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=1)"))

Turnover_q1 <- ggbarplot(beta_q1, x= "Non_lethal", y= "Turnover",
                         color = "black",  width = 0.6, lwd=0.3,
                         facet.by = "DT", fill = "Non_lethal",
                         add = "mean_se") +
  labs(x= element_blank(), y = titulo1) +
  theme_gray() + theme(text = element_text (size = 13)) +
  theme(legend.position = "right",
```

```
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  scale_y_continuous(limits = c(0,1)) +
  theme(legend.position = "top") +
  scale_fill_manual(values=c("#E5DFD6","#DDE8E9"))

print(Turnover_q1)
```




```
###################################
## Calculate turnover ratio (q2) ##
###################################
q2 <- pair_dis(otutable, qvalue = 2, metric = "V")
q2_df <- reshape2::melt(q2$L1_VqN, varnames=c("Site1", "Site2"),
                        value.name = "turnover", na.rm=TRUE)

q2_df_meta <- q2_df %>% inner_join(metadata, by = c("Site1"="SampleID")) %>%
  inner_join(metadata, by = c("Site2" = "SampleID")) %>% mutate(
    Types = paste0(SampleType.x, "vs", SampleType.y))
#write.table(q2_df_meta, file="./TURNOVER_q2.txt", sep = "\t")

# RATIO OF ASVs TURNOVER AT q=1
beta_q2 <- read.csv("../Data/Inter_q2.csv", header = TRUE, check.names = F)
titulo2 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=2)"))

Turnover_q2 <- ggbarplot(beta_q2, x= "Non_lethal", y= "Turnover",
                         color = "black",  width = 0.6, lwd=0.3,
                         facet.by = "DT", fill = "Non_lethal",
                         add = "mean_se") +
```

```
  labs(x= element_blank(), y = titulo2) +
  theme_gray() + theme(text = element_text (size = 13)) +
  theme(legend.position = "right",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  scale_y_continuous(limits = c(0,1)) +
  theme(legend.position = "top") +
  scale_fill_manual(values=c("#E5DFD6","#DDE8E9"))

print(Turnover_q2)
```



Venn-Diagram

```
library(tidyverse)

# Core microbiota (50%)
cloaca_50 <- read.csv("../Data/Core_50_Cloaca.csv",
                      check.names = F)
feces_50 <- read.csv("../Data/Core_50_Feces.csv",
                     check.names = F)
rectum_50 <- read.csv("../Data/Core_50_Rectum.csv",
                      check.names = F)
intestine_50 <- read.csv("../Data/Core_50_SmallIntestine.csv",
                         check.names = F)
stomach_50 <- read.delim("Core_50_Stomach.csv",
                         check.names = F)
```

```r
# Create Venn Diagramm
library(VennDiagram)

venn.plot_50 <- venn.diagram(
  x = list(Cloaca = cloaca_50$OTUID,
           Feces = feces_50$OTUID,
           Rectum = rectum_50$OTUID,
           Intestine = intestine_50$OTUID,
           Stomach = stomach_50$OTUID),
  category.names = c(
    expression(bold("Cloaca")),
    expression(bold("Feces")),
    expression(bold("Rectum")),
    expression(bold("Small intestine")),
    expression(bold("Stomach"))),
  filename = "viendo_50.tiff",
  output = TRUE,
  height = 3000,
  width = 3000,
  resolution = 300,
  compression = "lzw",
  units = "px",
  lwd = 6,
  lty = "blank",
  fill = c("#888888", "#D55E00", "#44AA99", "#DDCC77", "#CC6677"),
  cex = 1.5,
  fontface = "bold",
  fontfamily = "sans",
  cat.cex = 2,
  cat.fontface = "bold",
  cat.default.pos = "outer",
  cat.pos = c(-27, 27, 115, -125, -155),
  cat.dist = c(0.055, 0.055, 0.075, 0.060, 0.04),
  cat.fontfamily = "sans")
```

Beta diversity exploration

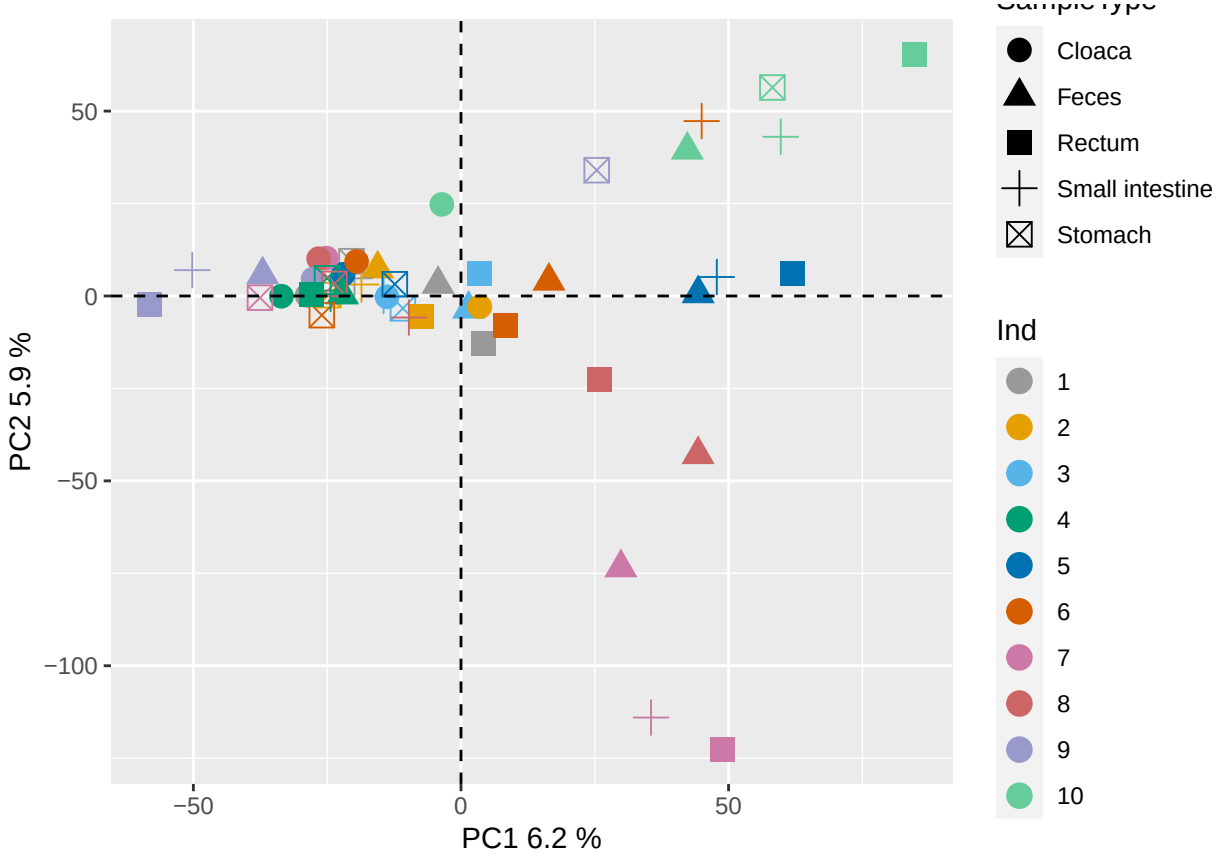
```r
## Loading libraries
#permanova
library(tidyverse)
library(compositions)
library(zCompositions)
library(CoDaSeq)

otutable <- read.csv("../Data/feature_table.csv", check.names = F, row.names = 1)
metadata <- read.csv("../Data/metadata.csv") %>% dplyr::select(SampleID:Ta)
taxonomy <- read.csv("../Data/taxonomy.csv", check.names = F)
metadata$Ind <- as.factor(metadata$Ind)
metadata$Library <- as.factor(metadata$Library)
metadata$SampleType <- as.factor(metadata$SampleType)
```

```
#### Transforming data "clr transformation/compositional data" ####
aldez.clr.transform <- aldex.clr(otutable, mc.samples = 999, denom = "all",
                                 verbose = FALSE, useMC = FALSE)
aldex.clr.transform.data <- t(getMonteCarloSample(aldez.clr.transform, 1))
```

```
# Labels to PCA axis
pcx.abund.aldex <- prcomp(aldex.clr.transform.data)

pc1 <- paste("PC1", round(sum(pcx.abund.aldex$sdev[1] ^2) / mvar(aldex.clr.transform.data) * 100, 1), "
pc2 <- paste("PC2", round(sum(pcx.abund.aldex$sdev[2] ^2) / mvar(aldex.clr.transform.data) * 100, 1), "

# Pallete color
paleta <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#0072B2",
            "#D55E00", "#CC79A7", "#CC6666", "#9999CC", "#66CC99")
print(paleta)
```

```
##  [1] "#999999" "#E69F00" "#56B4E9" "#009E73" "#0072B2" "#D55E00" "#CC79A7"
##  [8] "#CC6666" "#9999CC" "#66CC99"
```

```
# Create the base plot with only the arrows
pca_plot_aldex.clr <- ggplot() +
  theme_bw() +
  xlab(pc1) +
  ylab(pc2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setting theme
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "right") +
  theme_gray() +
  geom_point( #individuals
    data = data.frame(pcx.abund.aldex$x) %>%
      rownames_to_column(var = "SampleID") %>%
      left_join(metadata, by = "SampleID"),
    aes(x=PC1, y=PC2, shape = SampleType,
        color = Ind), size=4) +
  scale_color_manual(values = paleta) +
  geom_vline(xintercept = 0, linetype = 2) +
  geom_hline(yintercept = 0, linetype = 2)

print(pca_plot_aldex.clr)
```




```
#ggsave("pca_plot_aldex.clr.jpeg", width=5.5, height=5.5, dpi=300)

### PERMANOVA
set.seed(123)
library(vegan)
library(RVAideMemoire)
library(ggpubr)

meta_just <- data.frame(aldex.clr.transform.data, check.names = F) %>%
  rownames_to_column(var = "SampleID") %>%
  inner_join(metadata) %>% rename(SampleID="SampleID")

# perMANOVA incluyendo al individuo + sampletype
permanova_ind <- adonis2(aldex.clr.transform.data ~ SampleType,
                         data = meta_just, method = "euclidian",
                         permutations = 999)
permanova_ind
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = aldex.clr.transform.data ~ SampleType, data = meta_just, permutations = 999, metho
##            Df SumOfSqs      R2      F Pr(>F)
## SampleType  4    78891 0.09002 1.1129   0.02 *
## Residual   45   797458 0.90998
```

```
## Total      49   876349 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pairwise <- RVAideMemoire::pairwise.perm.manova(
  dist(aldex.clr.transform.data, method= "euclidian"),
  meta_just$SampleType, p.method = "BH", nperm = 999,  F = TRUE, R2 = TRUE)
pairwise
```

```
##
##  Pairwise comparisons using permutation MANOVAs on a distance matrix
##
## data:  dist(aldex.clr.transform.data, method = "euclidian") by meta_just$SampleType
## 999 permutations
##
##                 Cloaca Feces Rectum Small intestine
## Feces           0.010  -     -      -
## Rectum          0.010  0.980 -      -
## Small intestine 0.010  0.766 0.980  -
## Stomach         0.028  0.010 0.040  0.487
##
## P value adjustment method: BH
```

```
# BETADISPER ANALYSIS
matriz <- stats::dist(aldex.clr.transform.data, method = "euclidean")
mod <- betadisper(matriz, meta_just$SampleType)
pairwise_betadisper <- permutest(betadisper(matriz, meta_just$SampleType),
                                 pairwise = TRUE)
pairwise_betadisper
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups     4 4928.9 1232.22 9.0037    999  0.001 ***
## Residuals 45 6158.6  136.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                      Cloaca      Feces     Rectum Small intestine Stomach
## Cloaca                       1.0000e-03 1.0000e-03      2.6000e-02   0.167
## Feces           2.5099e-05              6.0000e-03      9.5100e-01   0.028
## Rectum          8.2518e-06 8.6947e-03                  6.4000e-02   0.001
## Small intestine 1.9256e-02 9.5070e-01 5.7177e-02                     0.189
## Stomach         1.4693e-01 2.7920e-02 5.2901e-04      1.8189e-01
```

Alpha diversity

Alpha taxonomic barplots

```
Hill <- read.csv(file = "../Data/feature_table.csv", header = TRUE, row.names = 1)

# Calcular la diversidad a los diferentes órdenes
library(hillR)
q0 <- hill_taxa(comm = t(Hill), q = 0)
q1 <- hill_taxa(comm = t(Hill), q = 1)
q2 <- hill_taxa(comm = t(Hill), q = 2)
Hill_div <- cbind(q0, q1, q2)
#write.table(Hill_div, file="./HillTaxa_Div.txt", sep = "\t")

## ALPHA DIVERSITY
library(tidyverse)
library(ggpubr)

#loading files
alpha <- Hill_div %>% as.data.frame() %>% rownames_to_column(var = "SampleID") %>%  dplyr::select(Sampl
metadata <- read.csv("../Data/metadata.csv", check.names = F) %>% dplyr::select(SampleID:Ta)
alpha <- alpha %>% inner_join(metadata)
```

```
shapiro.test(x =alpha$q0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q0
## W = 0.90708, p-value = 0.0008302
```

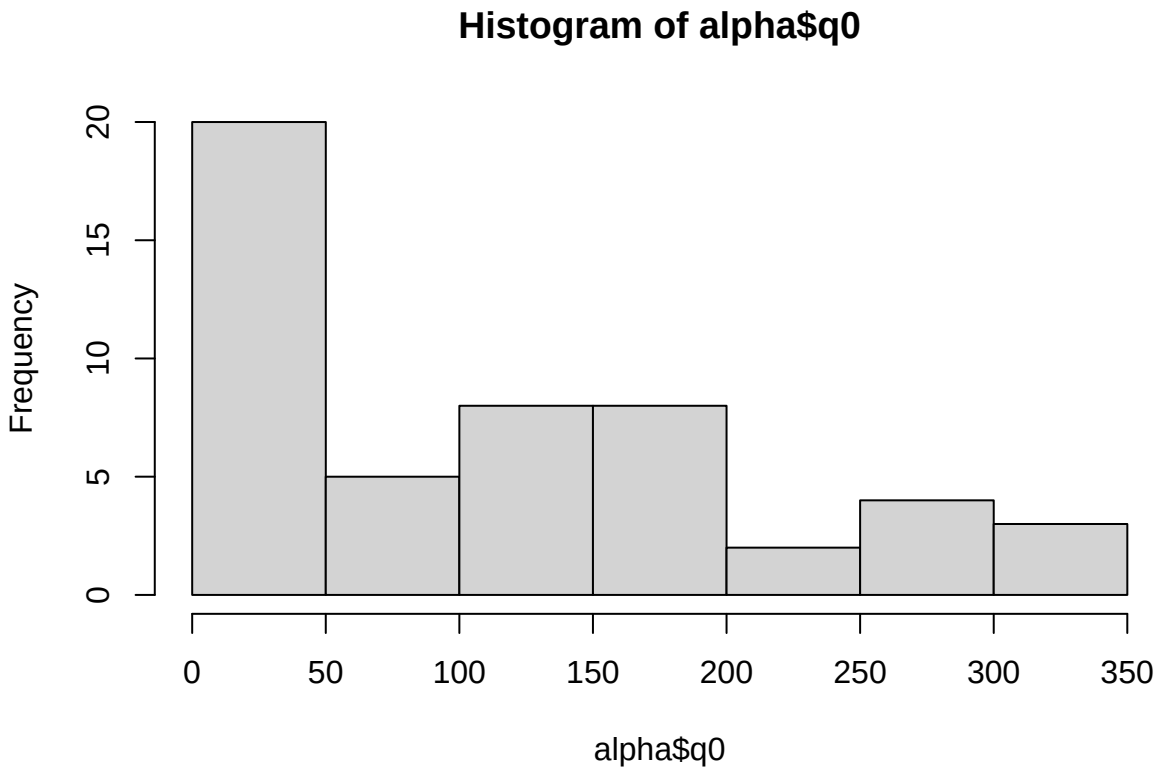
```
shapiro.test(x =alpha$q1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q1
## W = 0.89186, p-value = 0.0002615
```

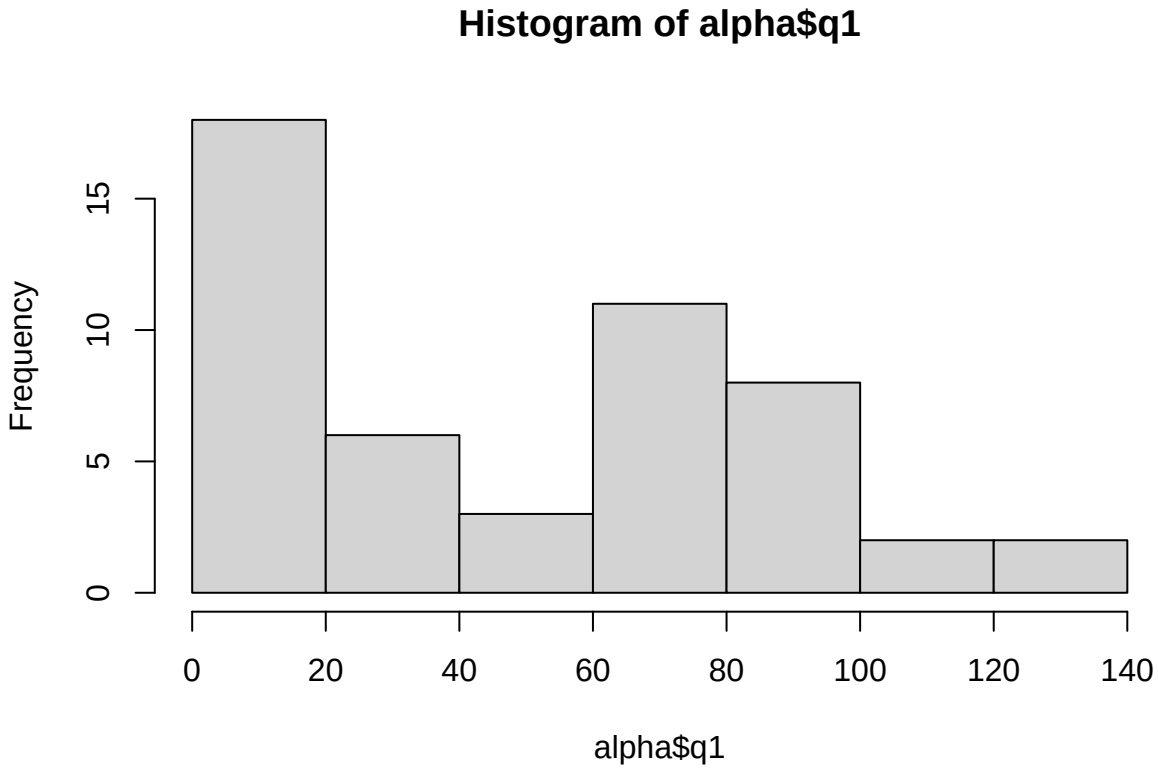
```
shapiro.test(x =alpha$q2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q2
## W = 0.90817, p-value = 0.0009042
```

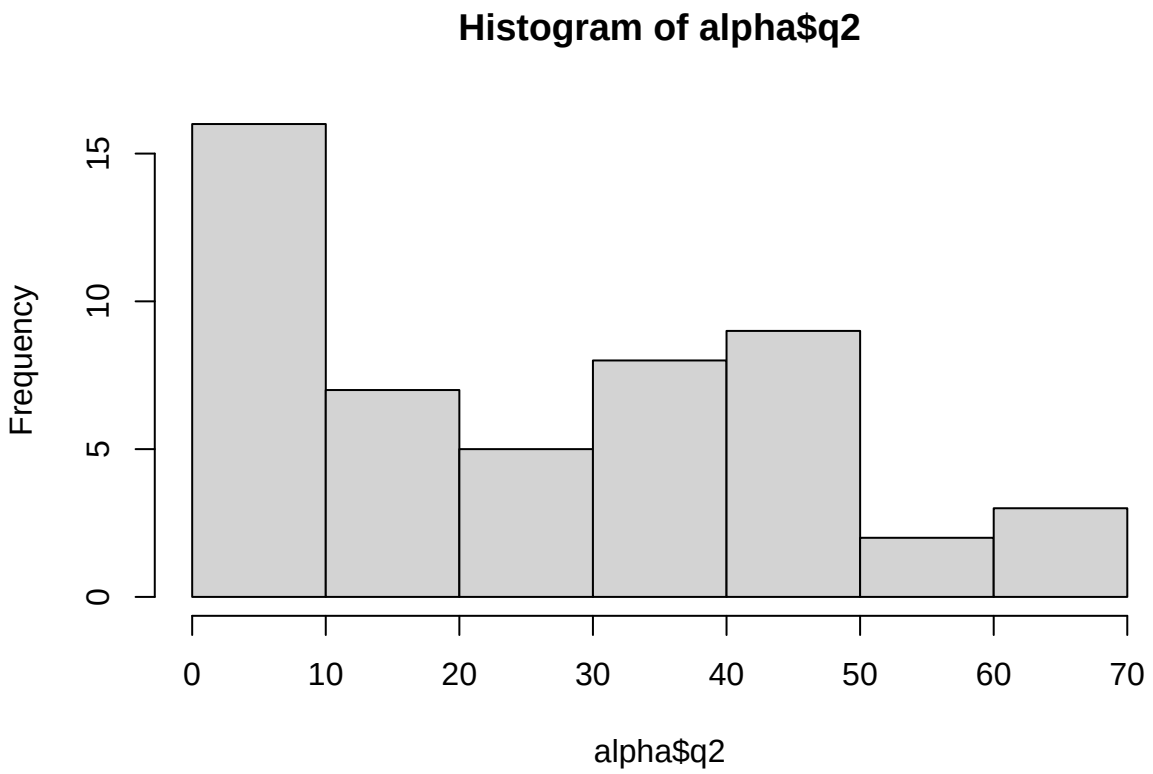
```
hist(alpha$q0)
```




```
hist(alpha$q1)
```

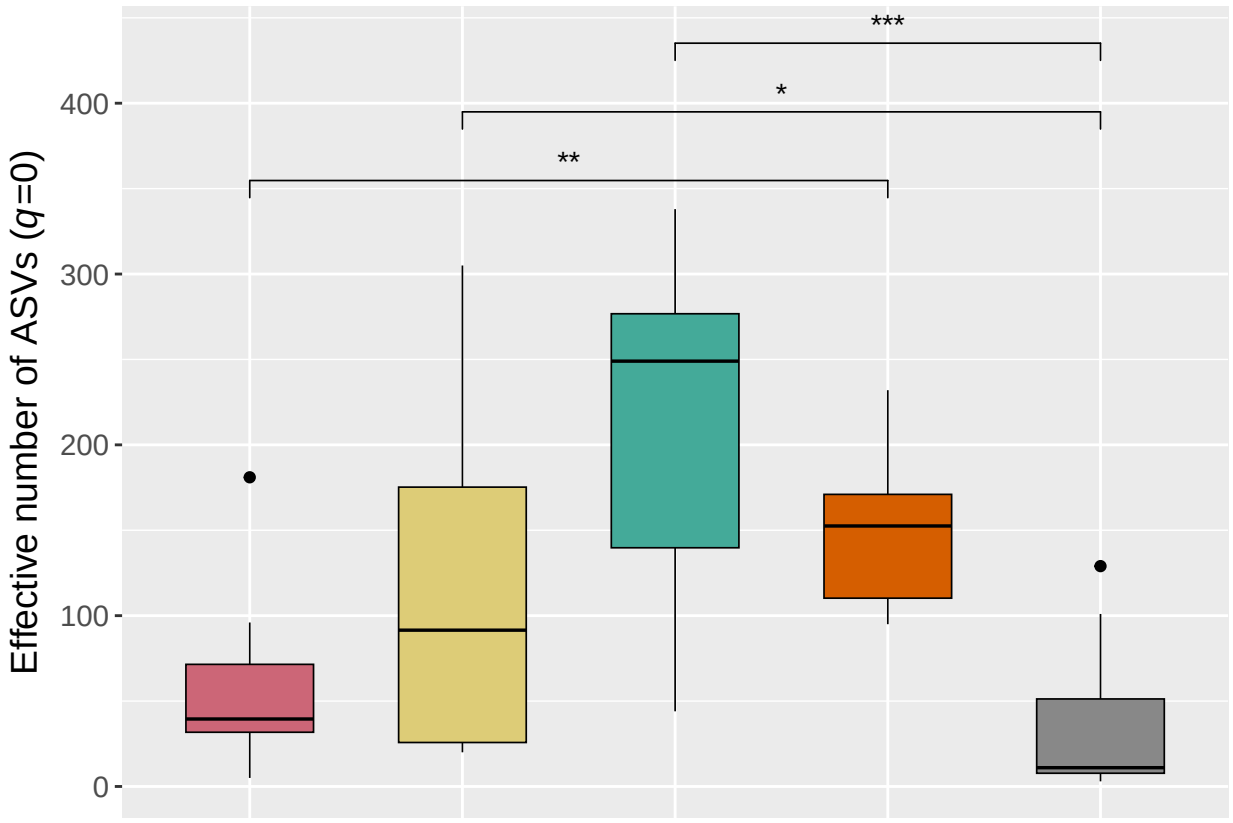



```
hist(alpha$q2)
```

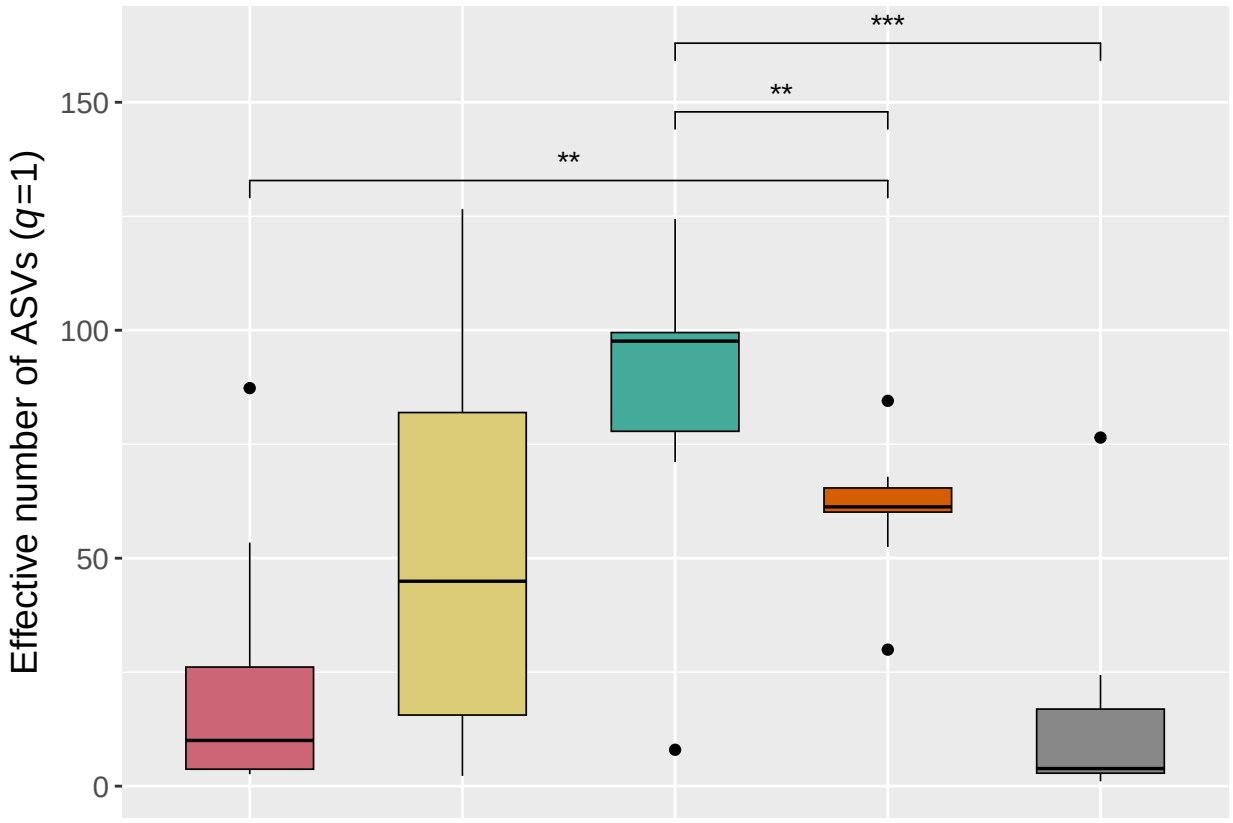



```
# Data are not normal

# Paired test (Wilcoxon) (q0)
feces <- subset(alpha, SampleType== "Feces", q0, drop = TRUE)
cloaca <- subset(alpha, SampleType== "Cloaca", q0, drop = TRUE)
stomach <-  subset(alpha, SampleType== "Stomach", q0, drop = TRUE)
intestine <- subset(alpha, SampleType== "Small intestine", q0, drop = TRUE)
rectum <-  subset(alpha, SampleType== "Rectum", q0, drop = TRUE)

FvsS_q0 <- wilcox.test(x= feces, y= stomach, paired = TRUE)
FvsI_q0 <- wilcox.test(x= feces, y= intestine, paired = TRUE)
FvsR_q0 <- wilcox.test(x= feces, y= rectum, paired = TRUE)
CvsS_q0 <- wilcox.test(x= cloaca, y= stomach, paired = TRUE)
CvsI_q0 <- wilcox.test(x= cloaca, y= intestine, paired = TRUE)
CvsR_q0 <- wilcox.test(x= cloaca, y= rectum, paired = TRUE)

# Paired test (Wilcoxon) (q1)
feces1 <- subset(alpha, SampleType== "Feces", q1, drop = TRUE)
cloaca1 <- subset(alpha, SampleType== "Cloaca", q1, drop = TRUE)
stomach1 <-  subset(alpha, SampleType== "Stomach", q1, drop = TRUE)
intestine1 <- subset(alpha, SampleType== "Small intestine", q1, drop = TRUE)
rectum1 <-  subset(alpha, SampleType== "Rectum", q1, drop = TRUE)

FvsS_q1 <- wilcox.test(x= feces1, y= stomach1, paired = TRUE)
FvsI_q1 <- wilcox.test(x= feces1, y= intestine1, paired = TRUE)
FvsR_q1 <- wilcox.test(x= feces1, y= rectum1, paired = TRUE)
CvsS_q1 <- wilcox.test(x= cloaca1, y= stomach1, paired = TRUE)
CvsI_q1 <- wilcox.test(x= cloaca1, y= intestine1, paired = TRUE)
CvsR_q1 <- wilcox.test(x= cloaca1, y= rectum1, paired = TRUE)
```

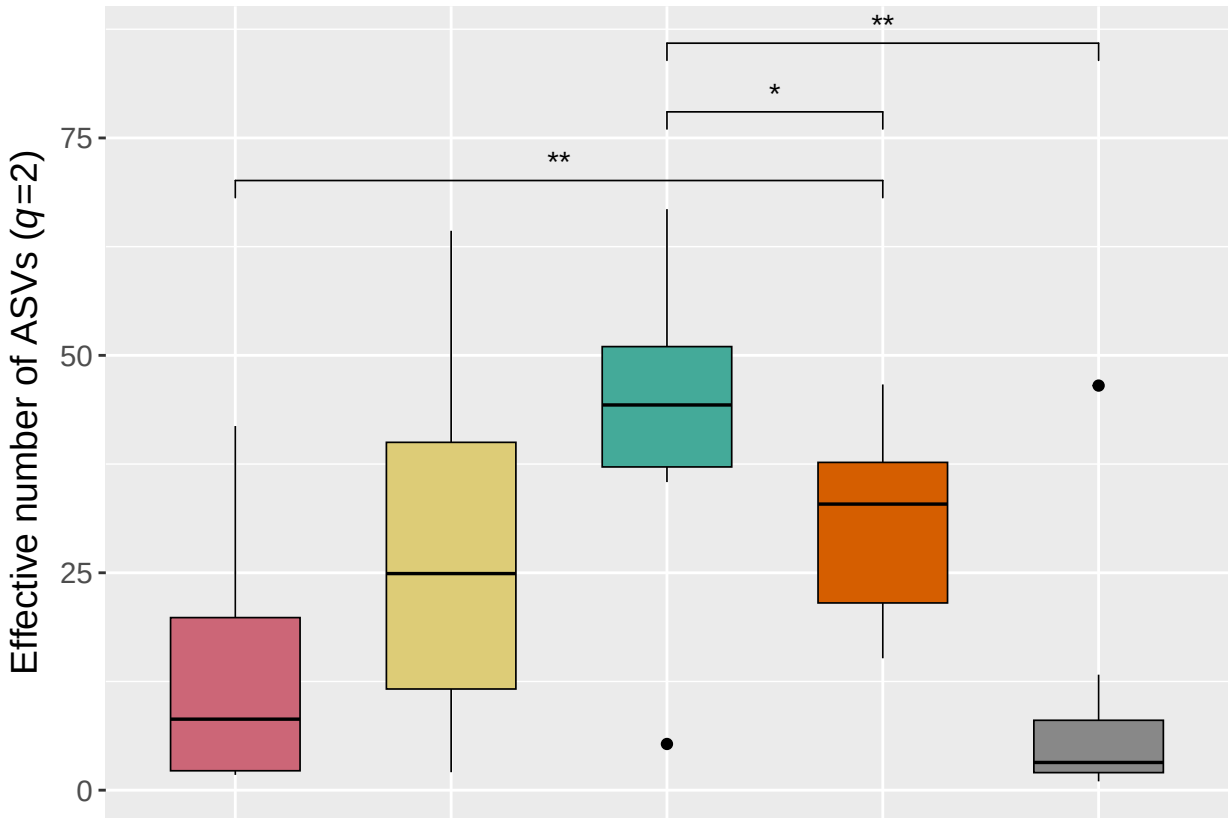
```r
# Paired test (Wilcoxon) (q2)
feces2 <- subset(alpha, SampleType== "Feces", q2, drop = TRUE)
cloaca2 <- subset(alpha, SampleType== "Cloaca", q2, drop = TRUE)
stomach2 <-  subset(alpha, SampleType== "Stomach", q2, drop = TRUE)
intestine2 <- subset(alpha, SampleType== "Small intestine", q2, drop = TRUE)
rectum2 <-  subset(alpha, SampleType== "Rectum", q2, drop = TRUE)

FvsS_q2 <- wilcox.test(x= feces2, y= stomach2, paired = TRUE)
FvsI_q2 <- wilcox.test(x= feces2, y= intestine2, paired = TRUE)
FvsR_q2 <- wilcox.test(x= feces2, y= rectum2, paired = TRUE)
CvsS_q2 <- wilcox.test(x= cloaca2, y= stomach2, paired = TRUE)
CvsI_q2 <- wilcox.test(x= cloaca2, y= intestine2, paired = TRUE)
CvsR_q2 <- wilcox.test(x= cloaca2, y= rectum2, paired = TRUE)

## ORIGINAL PAPER ##
#Visualize: Specify the comparisons you want with paired tests

# Diversity order q=0
my_comparisons_q0 <- list(c("Feces", "Stomach"),
                          c("Cloaca", "Small intestine"),
                          c("Cloaca", "Rectum"))

# Crear un argumento para dejar la (q) en italica.
titulo0 <- expression(paste("Effective number of ASVs (", italic("q"), "=0)"))
HillNumb_q0 <- ggboxplot(alpha, x= "SampleType", y= "q0",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"))+
  labs(x = element_blank(), y = titulo0) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  stat_compare_means(comparisons = my_comparisons_q0, label = "p.signif")+
  scale_fill_manual(values =   c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q0)
```



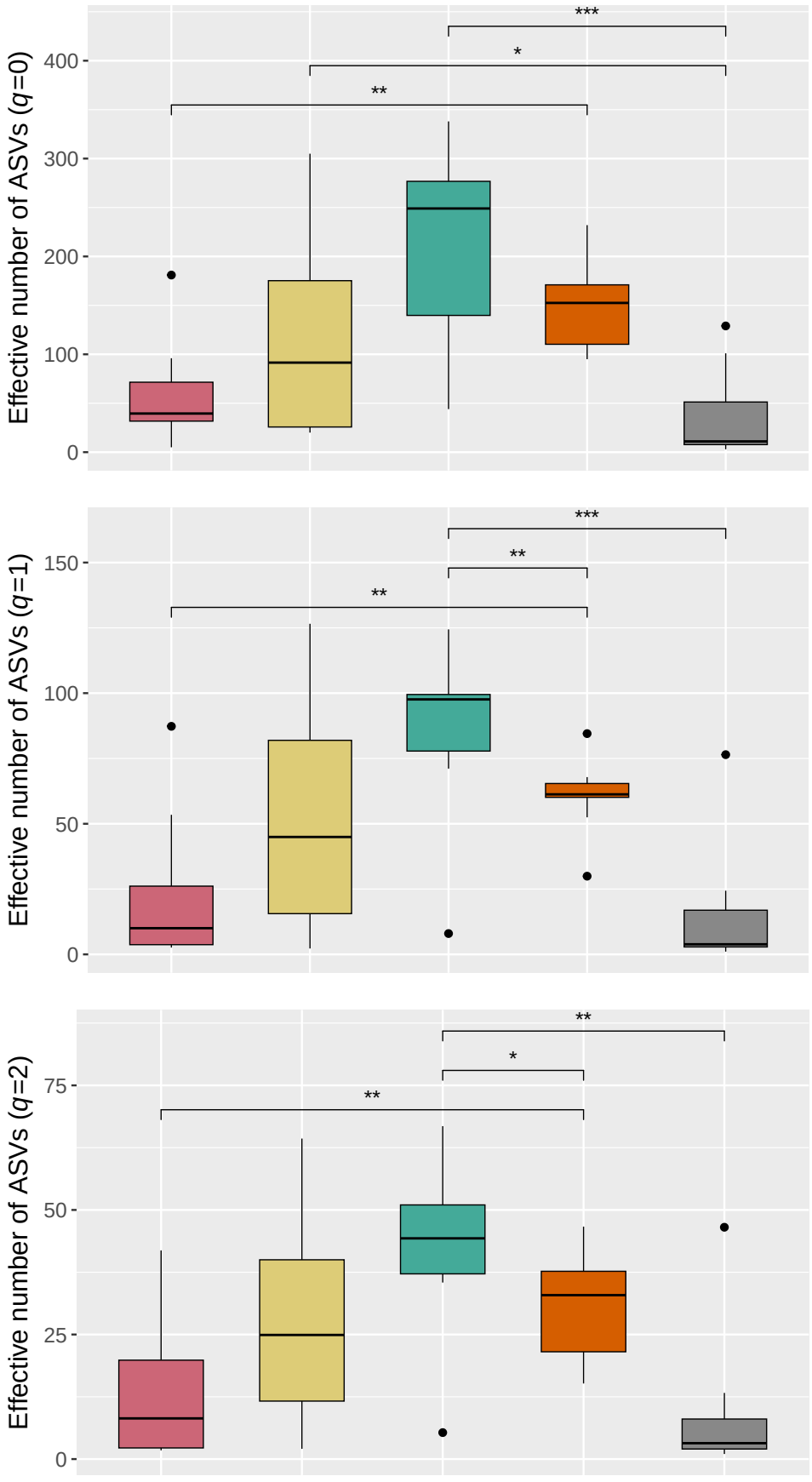

```
# Diversity order q=1
my_comparisons_q1 <- list(c("Feces", "Stomach"),
                          c("Feces", "Rectum"),
                          c("Cloaca", "Rectum"))

titulo1 <- expression(paste("Effective number of ASVs (", italic("q"), "=1)"))
HillNumb_q1 <- ggboxplot(alpha, x= "SampleType", y= "q1",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"))+
  labs(x = element_blank(), y = titulo1) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  stat_compare_means(comparisons = my_comparisons_q1, label = "p.signif")+
  scale_fill_manual(values =   c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q1)
```




```
# Diversity order q=2
my_comparisons_q2 <- list(c("Feces", "Stomach"),
                          c("Feces", "Rectum"),
                          c("Cloaca", "Rectum"))

titulo2 <- expression(paste("Effective number of ASVs (", italic("q"), "=2)"))
HillNumb_q2 <- ggboxplot(alpha, x= "SampleType", y= "q2",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"))+
  labs(x = element_blank(), y = titulo2) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  stat_compare_means(comparisons = my_comparisons_q2, label = "p.signif")+
  scale_fill_manual(values =   c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q2)
```




```
library(cowplot)
Graphics_boxplot_final <- plot_grid(HillNumb_q0, HillNumb_q1, HillNumb_q2,
                                    nrow = 3, ncol = 1,
                                    label_size = 13, rel_heights = c(1, 1, 1))
print(Graphics_boxplot_final)
```


Effective number of ASVs (q=0)
400
300
200
100
0
***
*
**
Effective number of ASVs (q=1)
150
100
50
0
***
**
**
Effective number of ASVs (q=2)
75
50
25
0
**
*
**

```
#ggsave("Graphics_boxplot_final.jpeg", width=6.0, height=11.0, dpi=300)
```

Alpha functional barplots

```
library(qiime2R)
funciones <- read_tsv("../Data/EC_predicted.tsv") %>%
  arrange(desc(sequence)) %>% column_to_rownames(var = "sequence")
tabla <- data.frame(read_q2biom("../Data/feature-table.biom")) %>%
  rownames_to_column(var = "sequence") %>% arrange(desc(sequence)) %>%
  column_to_rownames(  var = "sequence") %>%t() %>% as.data.frame()

# Run functional diversity by Hill numbers (at different q orders)
func_q0 <- hill_func(comm = tabla, traits = funciones, q = 0)
func_q1 <- hill_func(comm = tabla, traits = funciones, q = 1)
func_q2 <- hill_func(comm = tabla, traits = funciones, q = 2)

# Saved as table
Hill_Funct <- cbind(func_q0, func_q1, func_q2)
#write.table(func_q0, file="./hill_funct_q0.txt", sep = "\t")
#write.table(func_q1, file="./hill_funct_q1.txt", sep = "\t")
#write.table(func_q2, file="./hill_funct_q2.txt", sep = "\t")
```

```
## ALPHA FUNCTIONAL DIVERSITY
alpha <- read.csv("../Data/Functional_div.csv", header = TRUE, check.names = F)
Div_Funct <- alpha %>% inner_join(metadata, by = c("SampleID"="SampleID"))

# Normality test
shapiro.test(x = Div_Funct$MD_q0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Div_Funct$MD_q0
## W = 0.91336, p-value = 0.001368
```

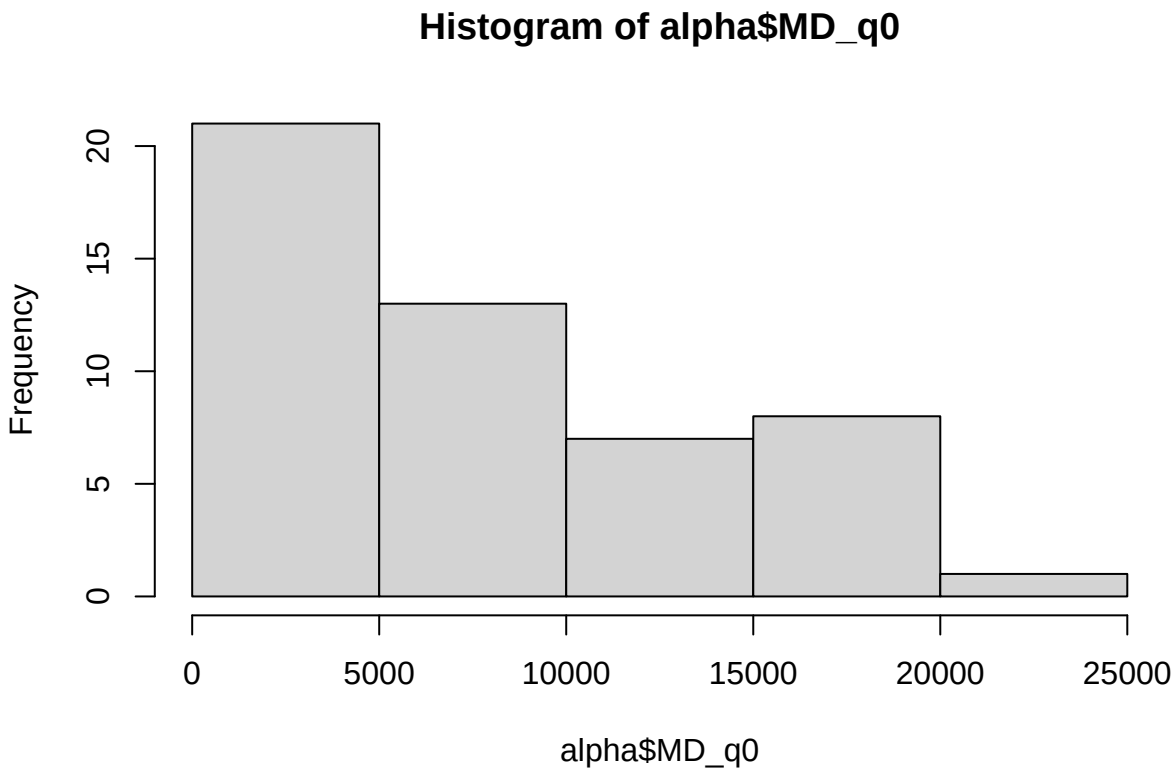
```
shapiro.test(x = Div_Funct$MD_q1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Div_Funct$MD_q1
## W = 0.92394, p-value = 0.003283
```

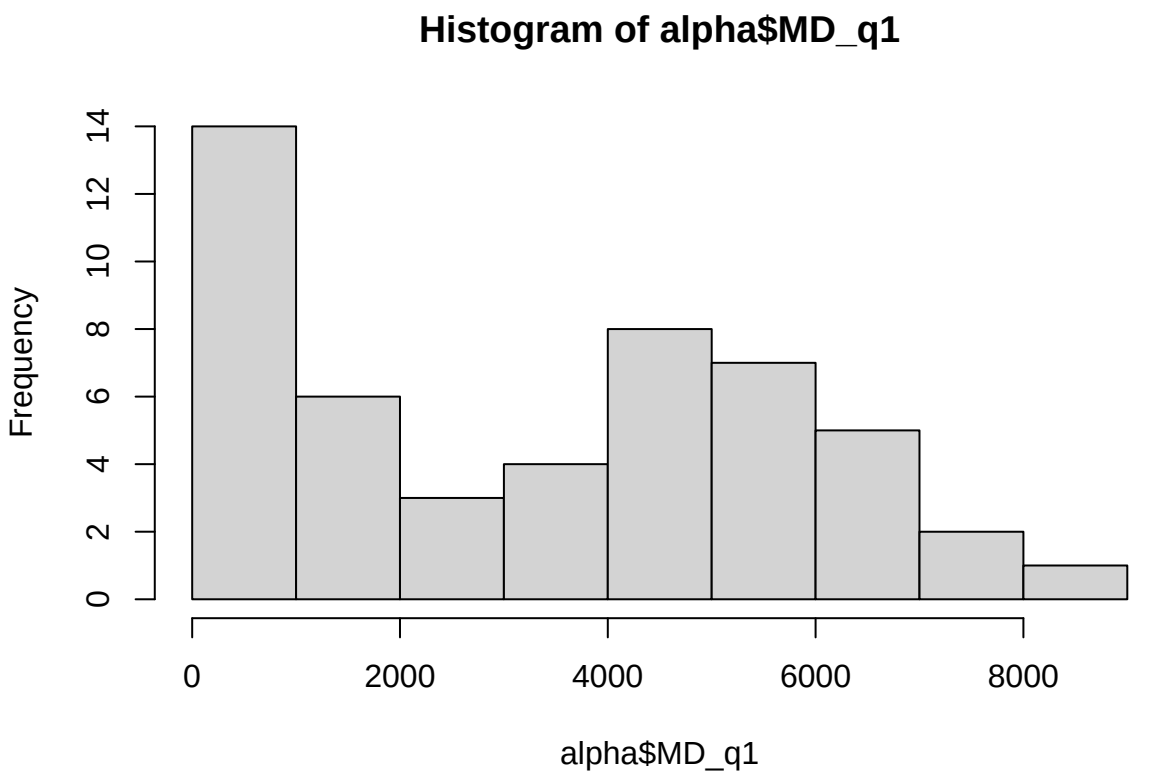
```
shapiro.test(x = Div_Funct$MD_q2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Div_Funct$MD_q2
## W = 0.94202, p-value = 0.01621
```

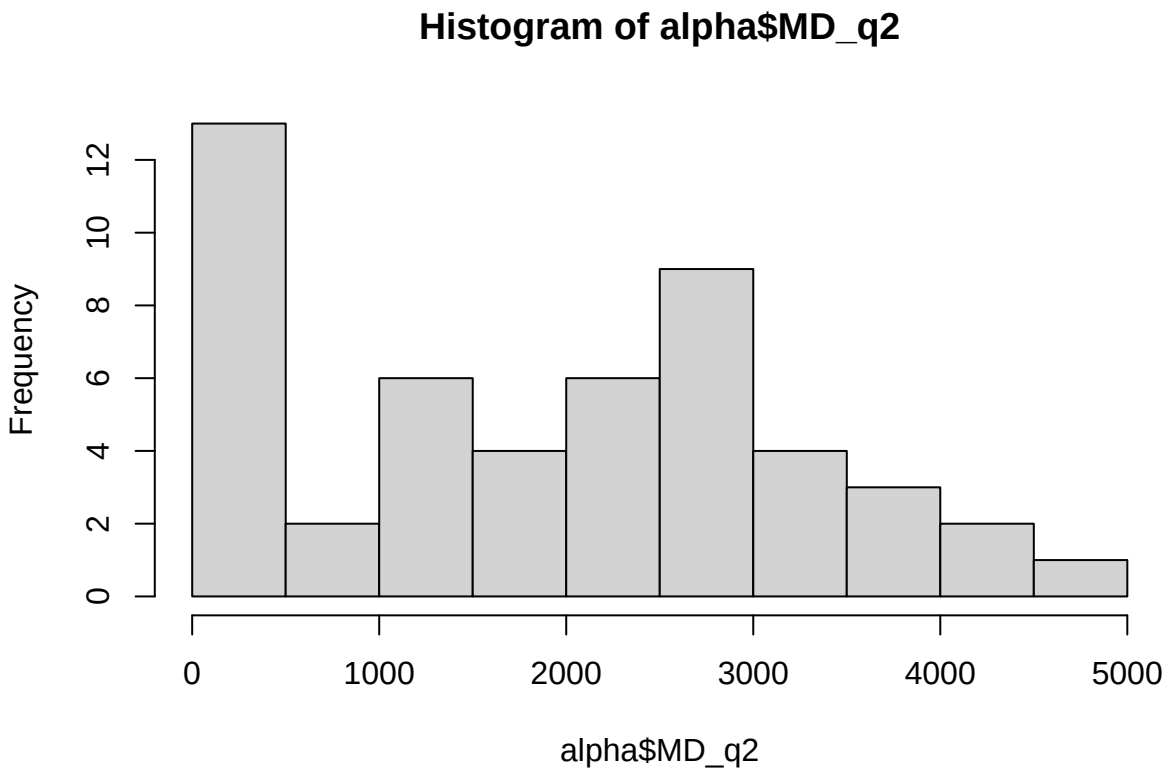
```
hist(alpha$MD_q0)
```




```
hist(alpha$MD_q1)
```

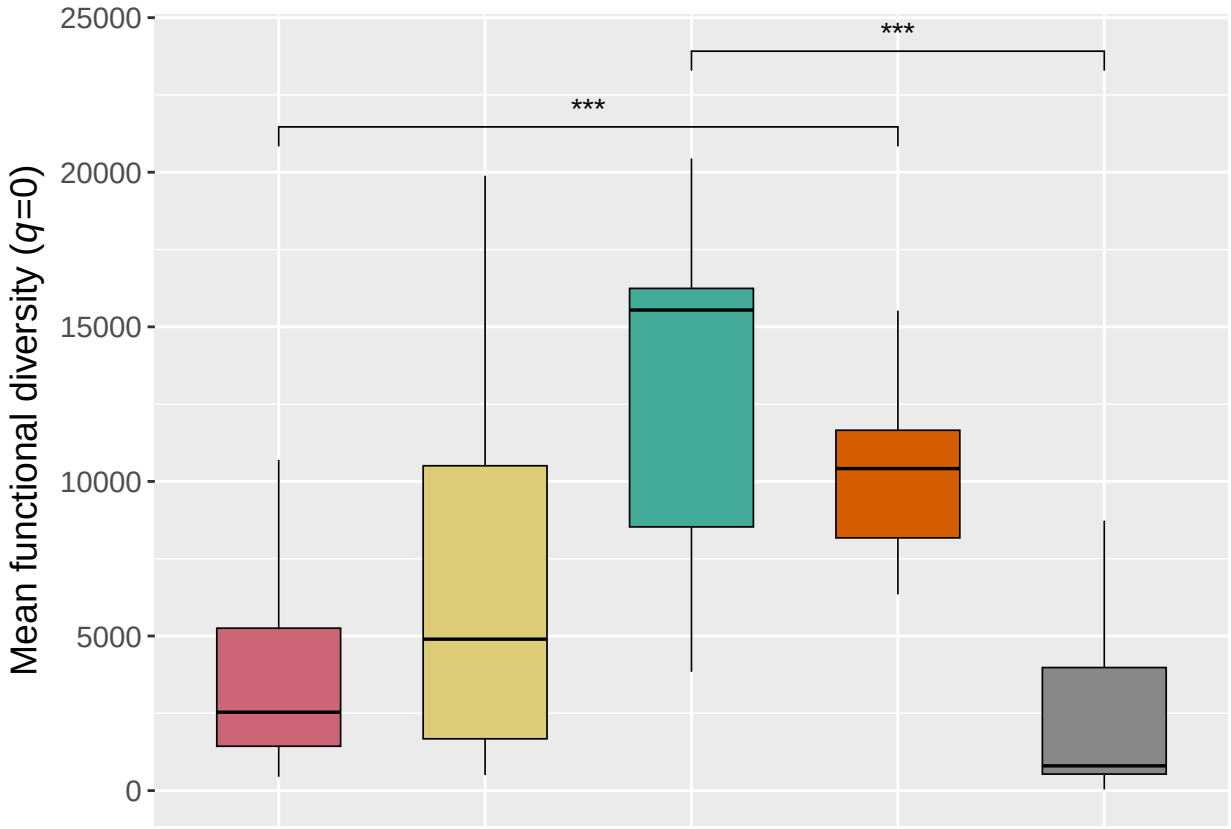


```
hist(alpha$MD_q2)
```

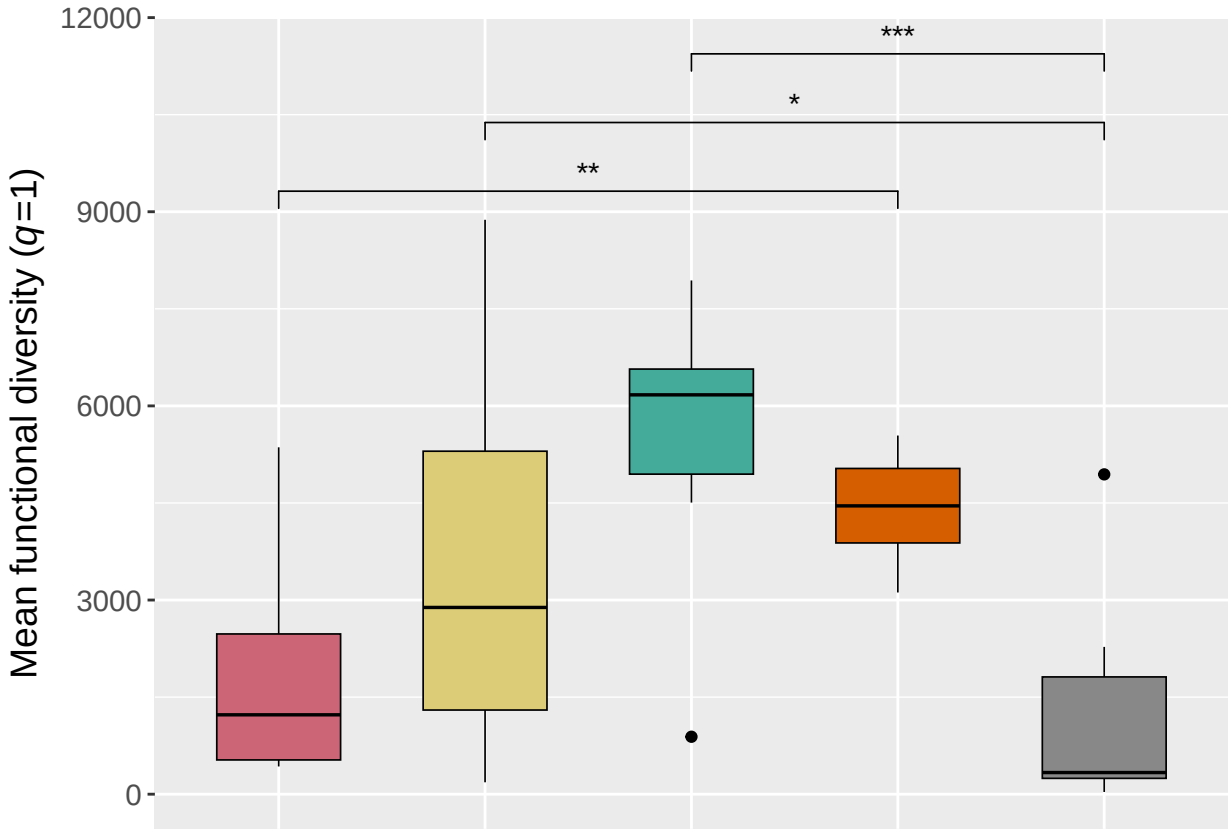



```
# Data are not normal

# Paired test (Wilcoxon) (q0)
feces <- subset(Div_Funct, SampleType== "Feces", MD_q0, drop = TRUE)
cloaca <- subset(Div_Funct, SampleType== "Cloaca", MD_q0, drop = TRUE)
stomach <-  subset(Div_Funct, SampleType== "Stomach", MD_q0, drop = TRUE)
intestine <- subset(Div_Funct, SampleType== "Small intestine", MD_q0, drop = TRUE)
rectum <-  subset(Div_Funct, SampleType== "Rectum", MD_q0, drop = TRUE)


FvsS_q0 <- wilcox.test(x= feces, y= stomach, paired = TRUE)
FvsI_q0 <- wilcox.test(x= feces, y= intestine, paired = TRUE)
FvsR_q0 <- wilcox.test(x= feces, y= rectum, paired = TRUE)
CvsS_q0 <- wilcox.test(x= cloaca, y= stomach, paired = TRUE)
CvsI_q0 <- wilcox.test(x= cloaca, y= intestine, paired = TRUE)
CvsR_q0 <- wilcox.test(x= cloaca, y= rectum, paired = TRUE)

# Paired test (Wilcoxon) (q1)
feces1 <- subset(Div_Funct, SampleType== "Feces", MD_q1, drop = TRUE)
cloaca1 <- subset(Div_Funct, SampleType== "Cloaca", MD_q1, drop = TRUE)
stomach1 <-  subset(Div_Funct, SampleType== "Stomach", MD_q1, drop = TRUE)
intestine1 <- subset(Div_Funct, SampleType== "Small intestine", MD_q1, drop = TRUE)
rectum1 <-  subset(Div_Funct, SampleType== "Rectum", MD_q1, drop = TRUE)

FvsS_q1 <- wilcox.test(x= feces1, y= stomach1, paired = TRUE)
FvsI_q1 <- wilcox.test(x= feces1, y= intestine1, paired = TRUE)
FvsR_q1 <- wilcox.test(x= feces1, y= rectum1, paired = TRUE)
```

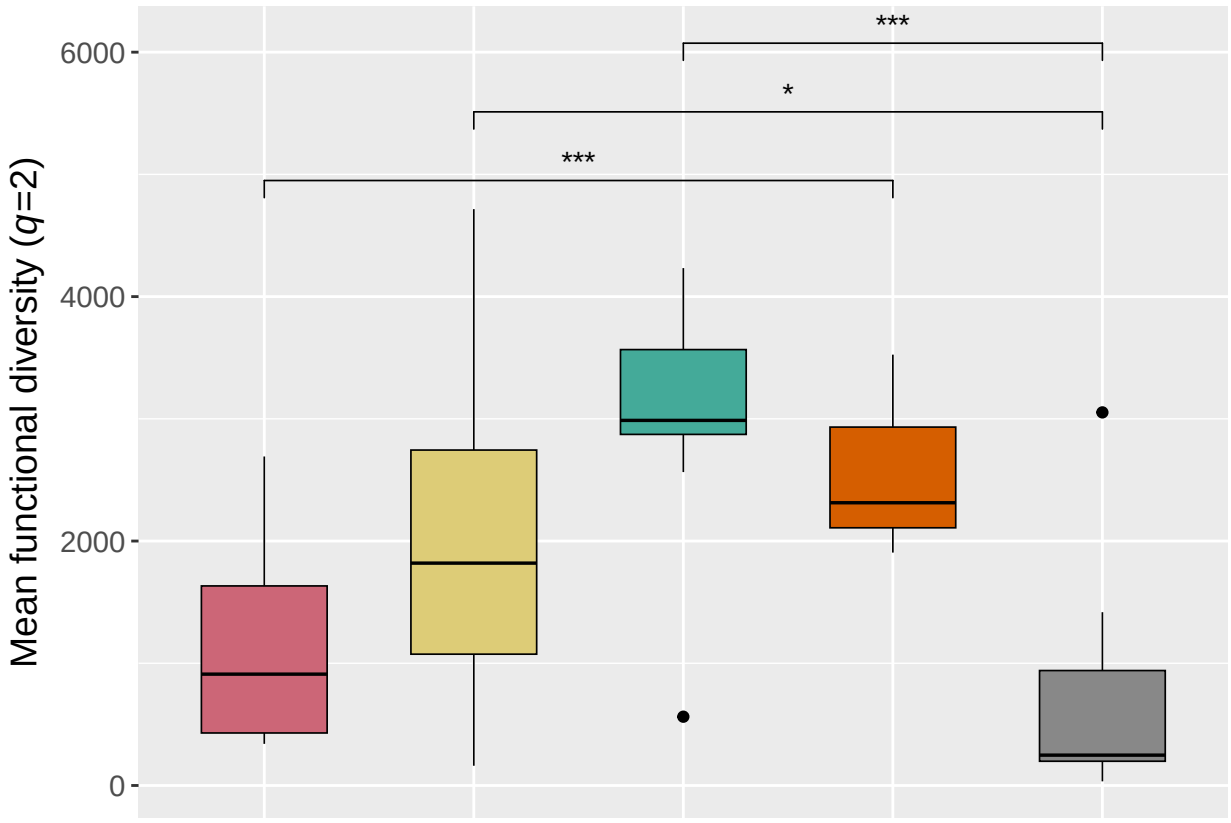
```
CvsS_q1 <- wilcox.test(x= cloaca1, y= stomach1, paired = TRUE)
CvsI_q1 <- wilcox.test(x= cloaca1, y= intestine1, paired = TRUE)
CvsR_q1 <- wilcox.test(x= cloaca1, y= rectum1, paired = TRUE)

# Paired test (Wilcoxon) (q2)
feces2 <- subset(Div_Funct, SampleType== "Feces", MD_q2, drop = TRUE)
cloaca2 <- subset(Div_Funct, SampleType== "Cloaca", MD_q2, drop = TRUE)
stomach2 <-  subset(Div_Funct, SampleType== "Stomach", MD_q2, drop = TRUE)
intestine2 <- subset(Div_Funct, SampleType== "Small intestine", MD_q2, drop = TRUE)
rectum2 <-  subset(Div_Funct, SampleType== "Rectum", MD_q2, drop = TRUE)

FvsS_q2 <- wilcox.test(x= feces2, y= stomach2, paired = TRUE)
FvsI_q2 <- wilcox.test(x= feces2, y= intestine2, paired = TRUE)
FvsR_q2 <- wilcox.test(x= feces2, y= rectum2, paired = TRUE)
CvsS_q2 <- wilcox.test(x= cloaca2, y= stomach2, paired = TRUE)
CvsI_q2 <- wilcox.test(x= cloaca2, y= intestine2, paired = TRUE)
CvsR_q2 <- wilcox.test(x= cloaca2, y= rectum2, paired = TRUE)

## ORIGINAL PAPER ##
#Visualize: Specify the comparisons you want with paired tests

# Diversity order q=0
my_comparisons_q0 <- list(c("Feces", "Stomach"),
                          c("Cloaca", "Rectum"))

# Crear un argumento para dejar la (q) en italica.
tituloA <- expression(paste("Mean functional diversity (", italic("q"), "=0)"))
HillNumb_q0 <- ggboxplot(Div_Funct, x= "SampleType", y= "MD_q0",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca")) +
  labs(x = element_blank(), y = tituloA) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  stat_compare_means(comparisons = my_comparisons_q0, label = "p.signif")+
  scale_fill_manual(values =  c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q0)
```



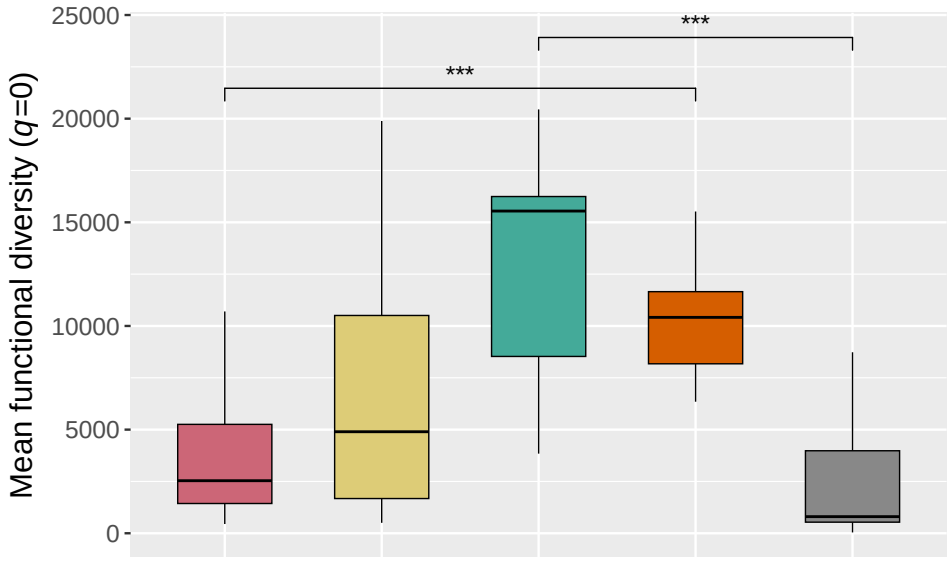

```
# Diversity order q=1
my_comparisons_q1 <- list(c("Feces", "Stomach"),
                          c("Cloaca", "Small intestine"),
                          c("Cloaca", "Rectum"))

tituloB <- expression(paste("Mean functional diversity (", italic("q"), "=1)"))
HillNumb_q1 <- ggboxplot(Div_Funct, x= "SampleType", y= "MD_q1",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca")) +
  labs(x = element_blank(), y = tituloB) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  stat_compare_means(comparisons = my_comparisons_q1, label = "p.signif")+
  scale_fill_manual(values =  c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q1)
```

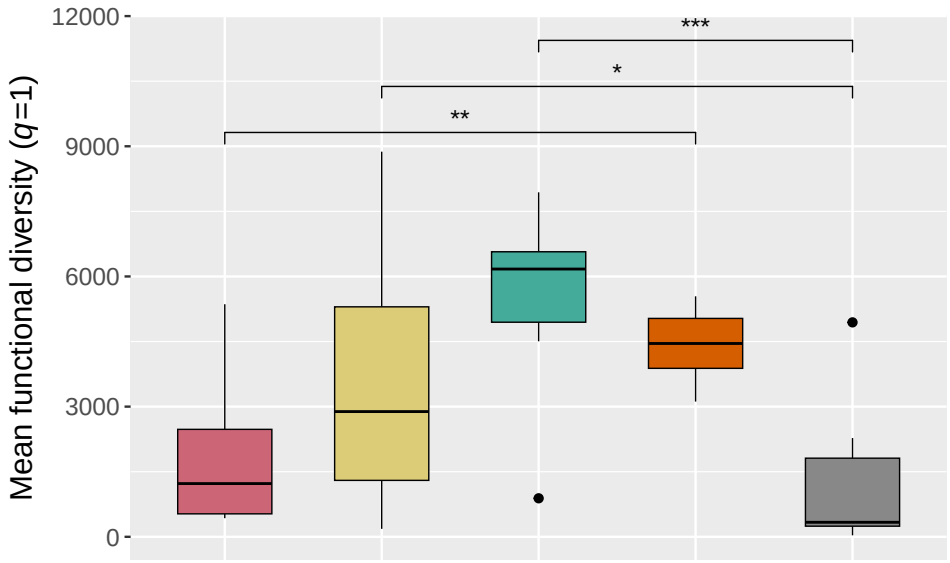


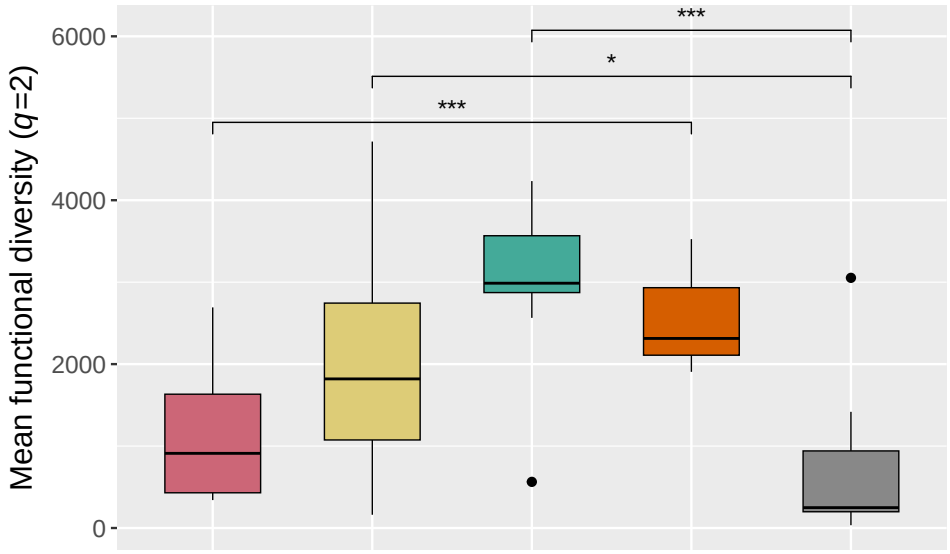

```
# Diversity order q=2
my_comparisons_q2 <- list(c("Feces", "Stomach"),
                          c("Cloaca", "Small intestine"),
                          c("Cloaca", "Rectum"))

tituloC <- expression(paste("Mean functional diversity (", italic("q"), "=2)"))
HillNumb_q2 <- ggboxplot(Div_Funct, x= "SampleType", y= "MD_q2",fill="SampleType",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca")) +
  labs(x = element_blank(), y = tituloC) +
  theme_gray() + theme(text = element_text (size = 14)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  stat_compare_means(comparisons = my_comparisons_q2, label = "p.signif")+
  scale_fill_manual(values =  c("#CC6677","#DDCC77","#44AA99", "#D55E00", "#888888"))
print(HillNumb_q2)
```




```
library(cowplot)
Boxplot_Final_FunctDiv <- plot_grid(HillNumb_q0, HillNumb_q1, HillNumb_q2,
                                    nrow = 3, ncol = 1,
                                    label_size = 13, rel_heights = c(1, 1, 1))
print(Boxplot_Final_FunctDiv)
```


Mean functional diversity (q=0)
25000
20000
15000
10000
5000
0
***
***


Mean functional diversity (q=1)
12000
9000
6000
3000
0
***
*
**


Mean functional diversity (q=2)
6000
4000
2000
0
***
*
***