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

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

12 - 01 - 2022

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:

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

```
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

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)

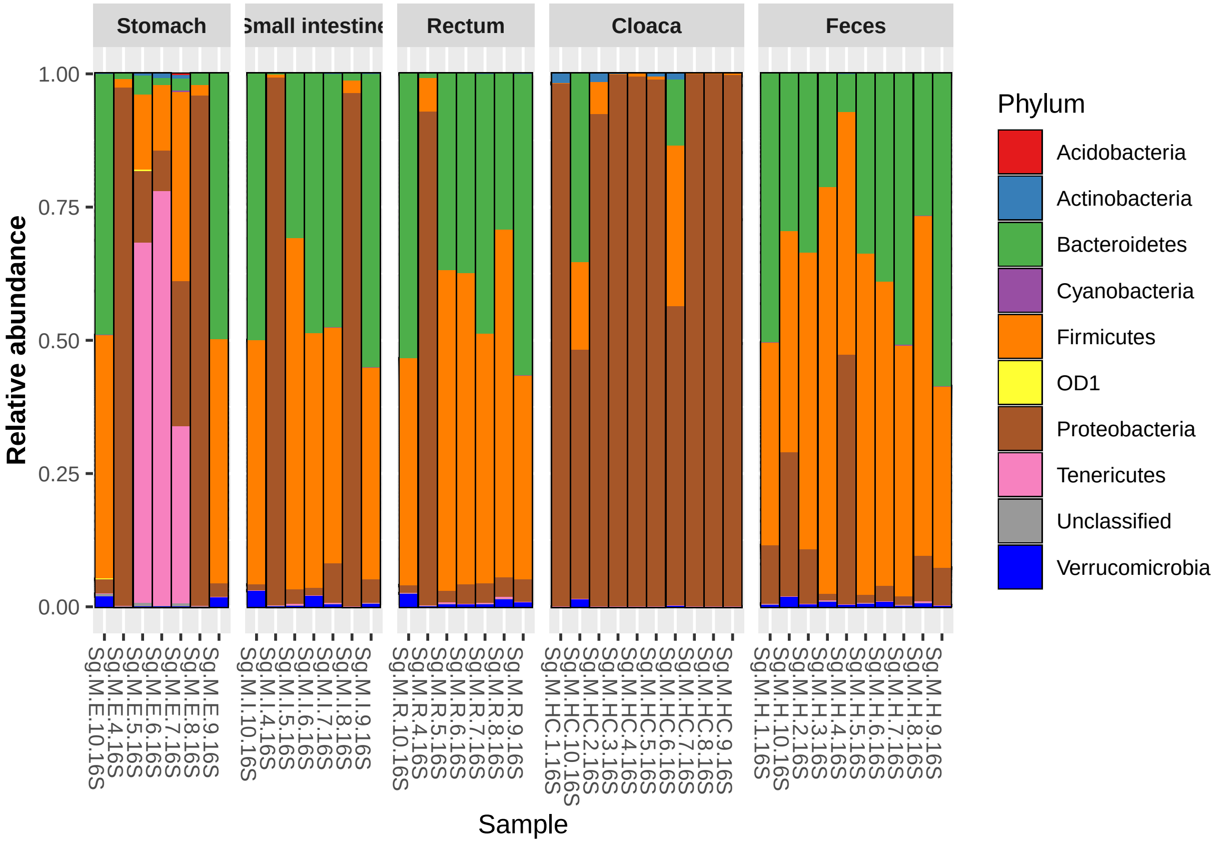
-o: output

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

Barplots

Phylum Barplot

```
#load packages and files
library(phyloseq)
library("DESeq2")
library(tidyverse)
library(RColorBrewer)

metadata <- read.csv(file = "../Data/Metadatos1.csv", header = TRUE, row.names = 1)
otu_table <- read.csv("../Data/otutable-taxonomy_ultima.csv", header = TRUE, row.names = 1)
taxonomy <- read.delim("../Data/taxonomy_ultima.txt", header = TRUE, row.names = 1)

# Create phyloseq object
SAM <- sample_data(metadata)
TAX <- tax_table(as.matrix(taxonomy))
OTU <- otu_table(otu_table, taxa_are_rows=TRUE)
physeq <- merge_phyloseq(OTU, TAX, SAM)

relative  = transform_sample_counts(physeq = physeq, function(OTU) OTU / sum(OTU))

# Filtering
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(brewer.pal(9, "Set1")[1:9], "blue")

Samples_DT_Phylum_grammicus <- plot_bar(
  physeq = relative, "Sample", fill = "Phylum")+
  facet_grid(~factor(SampleType, levels = c(
    "Stomach", "Small intestine", "Rectum", "Swab","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(
    strip.text.x = element_text(face = "bold"),
  axis.title.y = element_text(face = "bold")) +
  theme(text = element_text(size = 10))

print(Samples_DT_Phylum_grammicus)
```



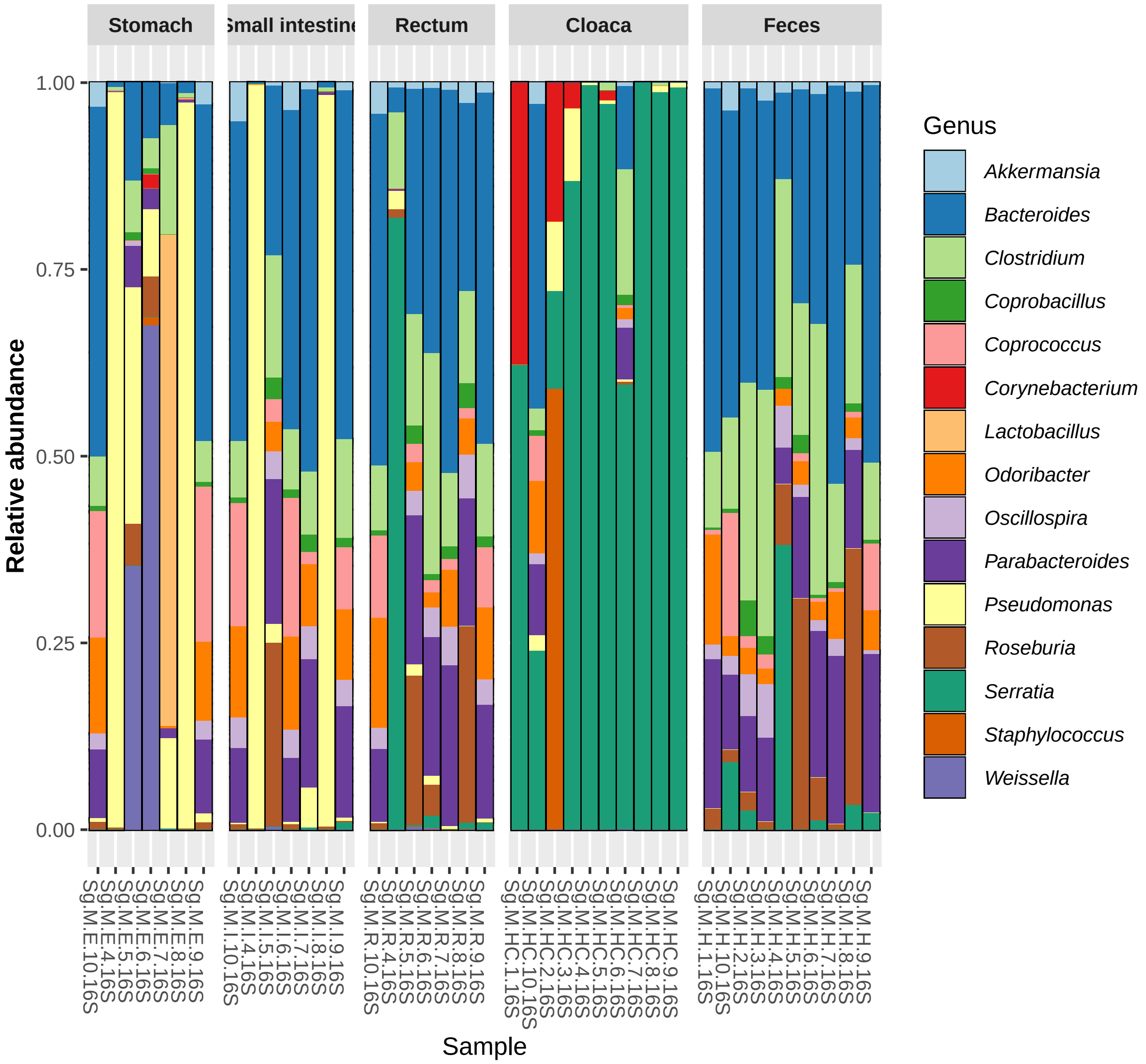
Genus Barplot

```
metadata <- read.csv(file = "../Data/Metadatos1.csv", header = TRUE,
                     row.names = 1) %>% mutate(
  SampleType=case_when(
    SampleType=="Swab"~"Cloaca",
    TRUE~as.character(SampleType)))

otu_table <- read.csv(file = "../Data/otutable-taxonomy_ultima.csv", check.names = F)
taxonomy <-  read.delim("../Data/taxonomy_ultima.txt", check.names = F) %>% mutate_at(
  c("Genus"), str_replace,"g__", "")
otutable_metadata <- otu_table  %>%   inner_join(taxonomy)

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=15) %>% rownames_to_column(
  var = "Genus") %>% filter(!Genus =="g__") %>% filter(
    !Genus =="Unassigned") %>% filter(
    !Genus =="g__[Clostridium]")%>% mutate_at(
      c("Genus"), str_replace,"g__", "")
list <- lista$Genus
```

```
taxonomy_filter <- taxonomy %>% filter(Genus %in% list)
taxonomy_1 <- taxonomy_filter %>% inner_join(otu_table, by =c(
  "#OTU ID"="#OTU ID")) %>% dplyr::select(1:8)

otu_table_1 <- read.csv(file = "../Data/otutable-taxonomy_ultima.csv", header = TRUE,
                        row.names = 1) %>% rownames_to_column(
                          var = "#OTU ID") %>% inner_join(
                            taxonomy_1, by = "#OTU ID") %>% dplyr::select(
                              -43:-49) %>% column_to_rownames(var = "#OTU ID")

taxo<- taxonomy_1 %>% column_to_rownames(var = "#OTU ID")

SAM <- sample_data(metadata)
TAX <- tax_table(as.matrix(taxo))
OTU <- otu_table(otu_table_1, taxa_are_rows=TRUE)
physeq <- merge_phyloseq(OTU, TAX, SAM)
relative  = transform_sample_counts(physeq = physeq, function(OTU) OTU / sum(OTU))

# filtering
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(brewer.pal(12, "Paired")[1:12], brewer.pal(8, "Dark2")[1:8])

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: Ignoring unknown parameters: res
```

```
print(Final_Genus_Sg)
```



Aldex GLM SG

```
#load packages and files
library(tidyverse)
library(compositions)
library(zCompositions)
library(CoDaSeq)

otutableA <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                        check.names = F, row.names = 1) %>% dplyr::select(-taxonomy)
otutableB <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                        check.names = F)
taxonomyA <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                        check.names = F) %>% dplyr::select(
  "#OTU ID", taxonomy)
```

```
ver<-otutableA %>% mutate(prom= rowSums(.)) %>% arrange(-prom)

metadataA <- read.csv("../Data/Metadatos1.csv", check.names = F)
metadataA$Ind<- as.factor(metadataA$Ind)
metadataA$Library<- as.factor(metadataA$Library)
metadataA$SampleType<- as.factor(metadataA$SampleType)

#transforming data
d.pro <- cmultRepl(t(otutableA), method="CZM", output="p-counts")
```

```
## No. corrected values:  771
```

```
d.clr.abund.codaseq<-codaSeq.clr(x = d.pro,samples.by.row = F)

meta_just<- data.frame(d.clr.abund.codaseq, check.names = F) %>% rownames_to_column(
  var = "SampleID") %>% inner_join(metadataA) %>%dplyr::select(SampleID,Ind, Library, SampleType)


meta_just$Ind<- as.factor(meta_just$Ind)
meta_just$Library<- as.factor(meta_just$Library)
meta_just$SampleType<- as.factor(meta_just$SampleType)


#otutableA %>% top_n(5)
var<- as.data.frame(d.clr.abund.codaseq) %>% dplyr::select(
  "0f4013b00115275df5e0ab6306716e8e") %>% rename(var="0f4013b00115275df5e0ab6306716e8e")
data_to_test<- var %>% rownames_to_column(var = "SampleID") %>% inner_join(meta_just)
#modelos
m1<- lm(var ~ Ind + SampleType, data_to_test)
m2<- lm(var~ SampleType, data_to_test )
library(lme4)
library(vegan)
m3<- lmer(var ~ SampleType + (1|Ind), data = data_to_test)
stats::anova(m1)
```

```
## Analysis of Variance Table
##
## Response: var
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Ind         9  82.988   9.221  4.3174  0.001469 **
## SampleType  4 233.789  58.447 27.3659 3.746e-09 ***
## Residuals  27  57.666   2.136
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
broom.mixed::glance(m3)
```

```
## # A tibble: 1 x 6
##   sigma logLik   AIC   BIC REMLcrit df.residual
##   <dbl>  <dbl> <dbl> <dbl>    <dbl>       <int>
## 1  1.45  -74.6  163.  175.     149.          34
```

```
broom::glance(m2)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic      p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>        <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.679         0.644  1.83      19.1 0.0000000169     4  -80.2  172.  183.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
otutableA <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                        check.names = F, row.names = 1) %>%
  dplyr::select(-taxonomy)#%>% column_to_rownames(var = "#OTU ID")

# FECES VS STOMACH
OTUT_FECES <- otutableA %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_STOMACH <- otutableA %>% 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 <- otutableA %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_INTESTINE <- otutableA %>% 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 <- otutableA %>% dplyr::select_at(vars(contains("M.H.")))
OTUT_RECTUM <- otutableA %>% 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 <- otutableA %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_STOMACH <- otutableA %>% 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 <- otutableA %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_INTESTINE <- otutableA %>% 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 <- otutableA %>% dplyr::select_at(vars(contains("M.HC.")))
OTUT_RECTUM <- otutableA %>% 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")
```

```
#load saved files
Feces_Stomach <- read.delim("../Data/ALDEXGLM_fECES_STOMACH.txt",
                            check.names = F, row.names = 1)
Feces_Intestine <- read.delim("../Data/ALDEXGLM_fECES_INTESTINE.txt",
                              check.names = F, row.names = 1)
```

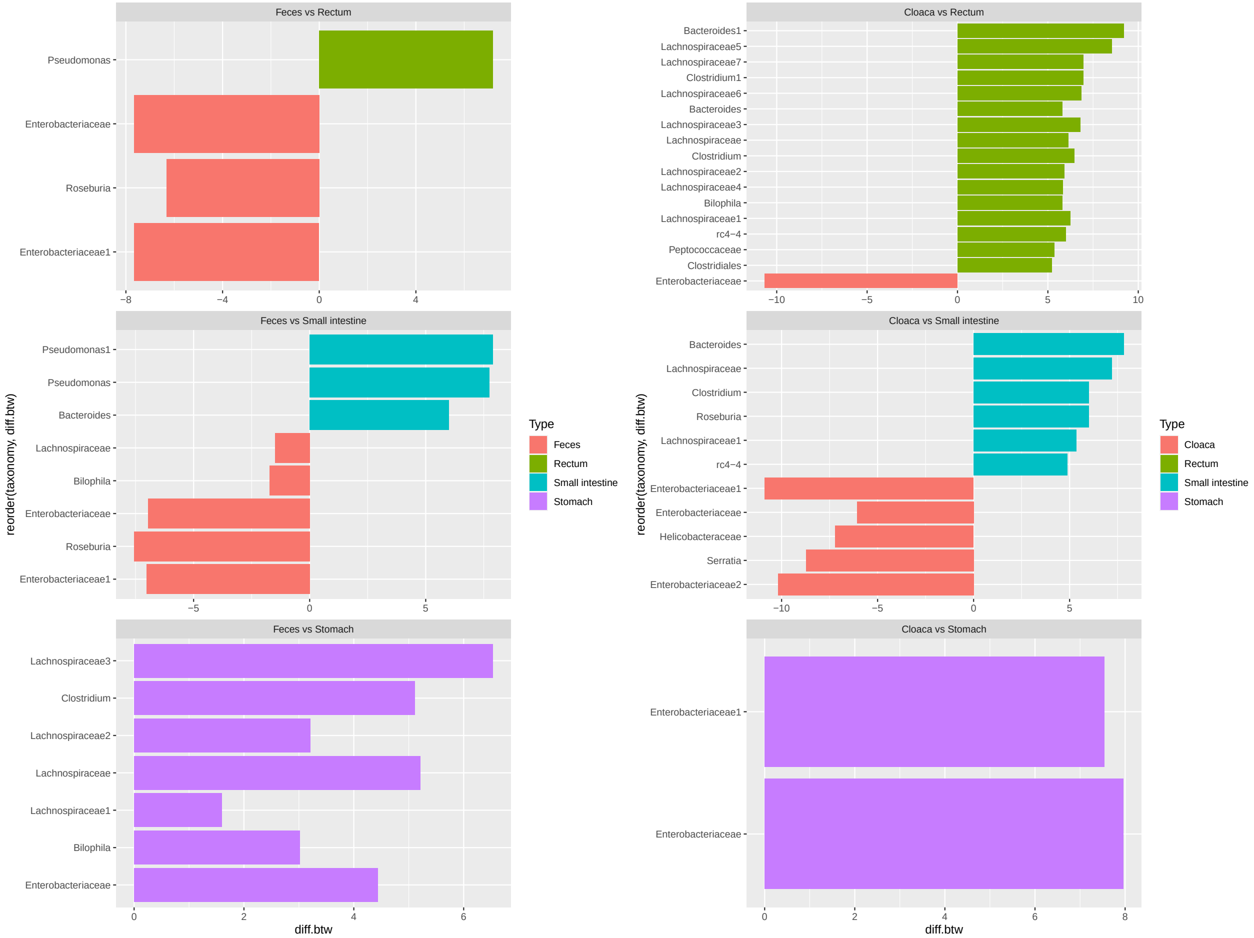
```
Feces_Rectum <- read.delim("../Data/ALDEXGLM_fECES_RECTUM.txt",
                           check.names = F, row.names = 1)
Cloaca_Stomach <- read.delim("../Data/ALDEXGLM_CLOACA_STOMACH.txt",
                             check.names = F, row.names = 1)
Cloaca_Intestine <- read.delim("../Data/ALDEXGLM_CLOACA_INTESTINE.txt",
                               check.names = F, row.names = 1)
Cloaca_Rectum <- read.delim("../Data/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))
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 = "#OTU ID")
aldex_table_FvsS<-  aldex_glm_FvsS %>% dplyr::select(
  pvalue="model.SampleTypeStomach Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% 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))
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 = "#OTU ID")
aldex_table_FvsI<-  aldex_glm_FvsI %>% dplyr::select(
  pvalue="model.SampleTypeSmall intestine Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% 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))
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 = "#OTU ID")
aldex_table_FvsR<-  aldex_glm_FvsR %>% dplyr::select(
  pvalue="model.SampleTypeRectum Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% inner_join(
      aldex_effect_FvsR_type)

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

### Cloaca versus Stomach ###
covar_CvsS<- metadata %>% filter(
  SampleType=="Swab"|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))
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 = "#OTU ID")
aldex_table_CvsS<-  aldex_glm_CvsS %>% dplyr::select(
  pvalue="model.SampleTypeStomach Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% 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=="Swab"|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))
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 = "#OTU ID")
aldex_table_CvsI<-  aldex_glm_CvsI %>% dplyr::select(
  pvalue="model.SampleTypeSmall intestine Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% inner_join(
      aldex_effect_CvsI_type)

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

### Cloaca versus Rectum ###
covar_CvsR<- metadata %>% filter(
  SampleType=="Swab"|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))
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 = "#OTU ID")
aldex_table_CvsR<-  aldex_glm_CvsR %>% dplyr::select(
  pvalue="model.SampleTypeRectum Pr(>|t|)") %>% filter(
    pvalue<0.05) %>% rownames_to_column(var = "#OTU ID")  %>% inner_join(
      taxonomyA)%>% inner_join(
      aldex_effect_CvsR_type)
```

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

```r
# Plot
library(cowplot)
GLMaldexFvsS <- read.delim("../Data/GLMaldexFvsS.txt", check.names = F)
GLMaldexFvsI <- read.delim("../Data/GLMaldexFvsI.txt", check.names = F)
GLMaldexFvsR <- read_tsv("../Data/GLMaldexFvsR.txt")

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")+facet_wrap(~Compare, ncol = 1, scales = "free")

GLMaldexCvsS <- read_tsv("../Data/GLMaldexCvsS.txt")
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") %>% rename(
    Other="Stomach")

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

C3<- GLMaldexCvsR  %>% mutate(Type = case_when(
  diff.btw >0  ~"Rectum",
  diff.btw<0 ~"Cloaca" )) %>% mutate(Compare="Cloaca vs Rectum")%>% 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")+facet_wrap(~Compare, ncol = 1, scales = "free")
```

```
a<-plot_grid(plot1,plot2)
a
```



```
#ggsave(plot = a, "plot_aldeglm.jpg", width = 16, height = 12)
```

Linear Regression

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

otutable <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                       row.names = 1) %>% dplyr::select(-taxonomy)
metadata <- read.csv("../Data/Metadatos1.csv", check.names = F)
metadata$Ind <- as.factor(metadata$Ind)
```

```
metadata$Library <- as.factor(metadata$Library)
metadata$SampleType <- as.factor(metadata$SampleType)
taxonomy <- read.delim("../Data/taxonomy_ultima.txt", check.names = F) %>% unite(
  taxa, Kingdom:Species, remove = F, sep = ";")

d.pro <- cmultRepl(t(otutable), method = "CZM", output = "p-counts")
```

```
## No. corrected values:  771
```

```
d.clr.abund.codaseq <- codaSeq.clr(x= d.pro, samples.by.row = F)
#clr_oject<- readRDS("clr_objetc.RDS")
phyl <- read_csv("../Data/level-2.csv")

phyl2 <- phyl %>% dplyr::select(index, contains("k__")) %>% column_to_rownames(
  var = "index")

d.pro <- cmultRepl(t(phyl2), method = "CZM", output = "p-counts")
```

```
## No. corrected values:  108
```

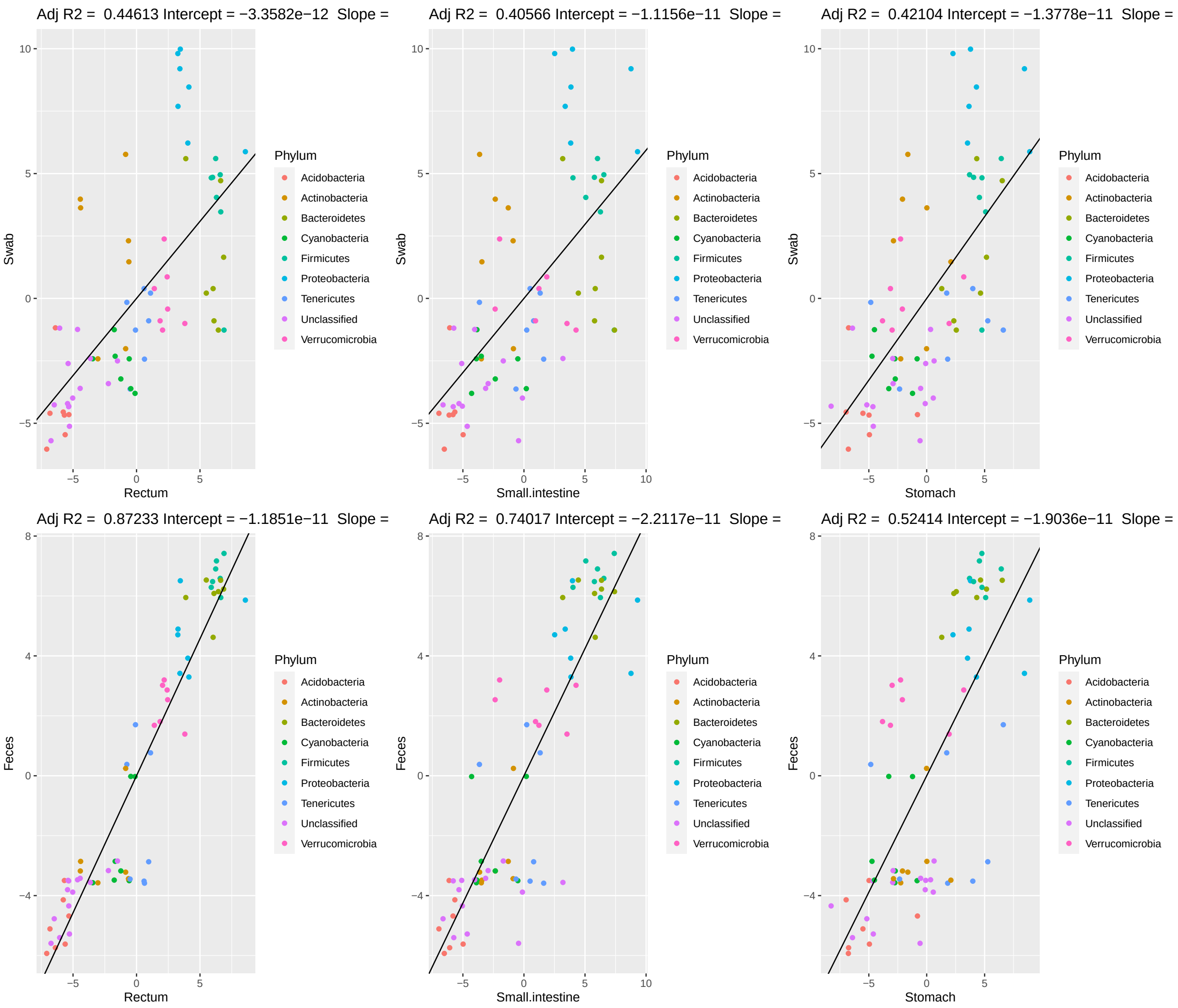
```
d.clr.abund.codaseq <- codaSeq.clr(x= d.pro, samples.by.row = F)
```

```
### Swab versus Rectum

phyl_S_R <- data.frame(t(d.clr.abund.codaseq))%>% rownames_to_column(
  var = "index") %>% inner_join(
  phyl) %>% dplyr::select(c(1:11), SampleType) %>% filter(
    SampleType=="Rectum"|SampleType=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("k__"),
               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, "ver.tsv")

SR <- read.csv("../Data/Swab_Rectum.csv")
data.lm_SR <- lm(Swab ~ Rectum, SR)
Swab_Rectum <- SR %>% ggplot(aes(x=Rectum, y=Swab, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_SR)[[2]], intercept = coef(data.lm_SR)[[1]])+
  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)))

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


### Swab versus Small intestine
```

```
phyl_S_I <- data.frame(t(d.clr.abund.codaseq))%>% rownames_to_column(
  var = "index") %>% inner_join(
  phyl) %>% dplyr::select(c(1:11), SampleType) %>% filter(
    SampleType=="Small intestine"|SampleType=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("k__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>% replace(is.na(.), 0)

otuSI <- phyl_S_I %>% dplyr::select(-index)
namesotuSI <- otuSI$names
#write_tsv(phyl_S_I, "Swab_Intestine.tsv")


SI <- read.csv("../Data/Swab_Intestine.csv")
data.lm_SI <- lm(Swab ~ Small.intestine, SI)
Swab_Small_intestine <- SI %>% ggplot(
  aes(x=Small.intestine, y=Swab, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_SI)[[2]], intercept = coef(data.lm_SI)[[1]])+
  labs(title = paste("Adj R2 = ",signif(summary(data.lm_SI)$adj.r.squared, 5),
                     "Intercept =",signif(data.lm_SI$coef[[1]],5 ),
                     " Slope =",signif(data.lm_SI$coef[[2]], 5),
                     " P =",signif(summary(data.lm_SI)$coef[2,4], 5)))



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

### Swab versus Stomach

phyl_S_St <- 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=="Swab") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("k__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>% replace(
                   is.na(.), 0)

otu_S_St <- phyl_S_St %>% dplyr::select(-index)
namesotuSSt <- otu_S_St$names
#write_tsv(phyl_S_St, "Swab_Stomach.tsv")

SSt <- read.csv("../Data/Swab_Stomach.csv")
data.lm_SSt <- lm(Swab ~ Stomach, SSt)
Swab_Stomach <- SSt %>% ggplot(aes(x=Stomach, y=Swab, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_SSt)[[2]], intercept = coef(data.lm_SSt)[[1]])+
  labs(title = paste("Adj R2 = ",signif(summary(data.lm_SSt)$adj.r.squared, 5),
                     "Intercept =",signif(data.lm_SSt$coef[[1]],5 ),
                     " Slope =",signif(data.lm_SSt$coef[[2]], 5),
                     " P =",signif(summary(data.lm_SSt)$coef[2,4], 5)))
```

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


### Feces versus Rectum

phyl_F_R <- data.frame(t(d.clr.abund.codaseq))%>% rownames_to_column(
  var = "index") %>% inner_join(
  phyl) %>% dplyr::select(c(1:11), SampleType) %>% filter(
    SampleType=="Rectum"|SampleType=="Feces") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("k__"),
               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)
namesotuFR <- otu_F_R$names
#write_tsv(phyl_F_R, "Feces_Rectum.tsv")

F_R <- read.csv("../Data/Feces_Rectum.csv")
data.lm_FR <- lm(Feces ~ Rectum, F_R)
Feces_Rectum <- F_R %>% ggplot(aes(x=Rectum, y=Feces, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_FR)[[2]], intercept = coef(data.lm_FR)[[1]])+
  labs(title = paste("Adj R2 = ",signif(summary(data.lm_FR)$adj.r.squared, 5),
                     "Intercept =",signif(data.lm_FR$coef[[1]],5 ),
                     " Slope =",signif(data.lm_FR$coef[[2]], 5),
                     " P =",signif(summary(data.lm_FR)$coef[2,4], 5)))


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

### Feces versus Small intestine

phyl_F_I <- data.frame(t(d.clr.abund.codaseq))%>% rownames_to_column(
  var = "index") %>% inner_join(
  phyl) %>% dplyr::select(c(1:11), SampleType) %>% filter(
    SampleType=="Small intestine"|SampleType=="Feces") %>%
  #dplyr::select(contains(c("HC", "R"))) %>%
  pivot_longer(cols = starts_with("k__"),
               names_to = "names", values_to = "values") %>% pivot_wider(
                 names_from = SampleType, values_from = values) %>% replace(
                   is.na(.), 0)

otu_F_I <- phyl_F_I %>% dplyr::select(-index)
namesotuFI <- otu_F_I $names
#write_tsv(phyl_F_I, "Feces_Small_intestine.tsv")

F_I <- read.csv("../Data/Feces_Small_intestine.csv")
data.lm_FI <- lm(Feces ~ Small.intestine, F_I)
Feces_Small_intestine <- F_I %>% ggplot(aes(
  x=Small.intestine, y=Feces, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_FI)[[2]], intercept = coef(data.lm_FI)[[1]])+
  labs(title = paste("Adj R2 = ",signif(summary(data.lm_FI)$adj.r.squared, 5),
                     "Intercept =",signif(data.lm_FI$coef[[1]],5 ),
                     " Slope =",signif(data.lm_FI$coef[[2]], 5),
                     " P =",signif(summary(data.lm_FI)$coef[2,4], 5)))

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

### Feces versus Stomach

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

otu_F_S <- phyl_F_S %>% dplyr::select(-index)
namesotuFS <- otu_F_S$names
#write_tsv(phyl_F_S, "Feces_Stomach.tsv")

F_S <- read.csv("../Data/Feces_Stomach.csv")
data.lm_FS <- lm(Feces ~ Stomach, F_S)
Feces_Stomach <- F_S %>% ggplot(aes(x=Stomach, y=Feces, color=Phylum)) + geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_abline(slope = coef(data.lm_FS)[[2]], intercept = coef(data.lm_FS)[[1]])+
  labs(title = paste("Adj R2 = ",signif(summary(data.lm_FS)$adj.r.squared, 5),
                     "Intercept =",signif(data.lm_FS$coef[[1]],5 ),
                     " Slope =",signif(data.lm_FS$coef[[2]], 5),
                     " P =",signif(summary(data.lm_FS)$coef[2,4], 5)))

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

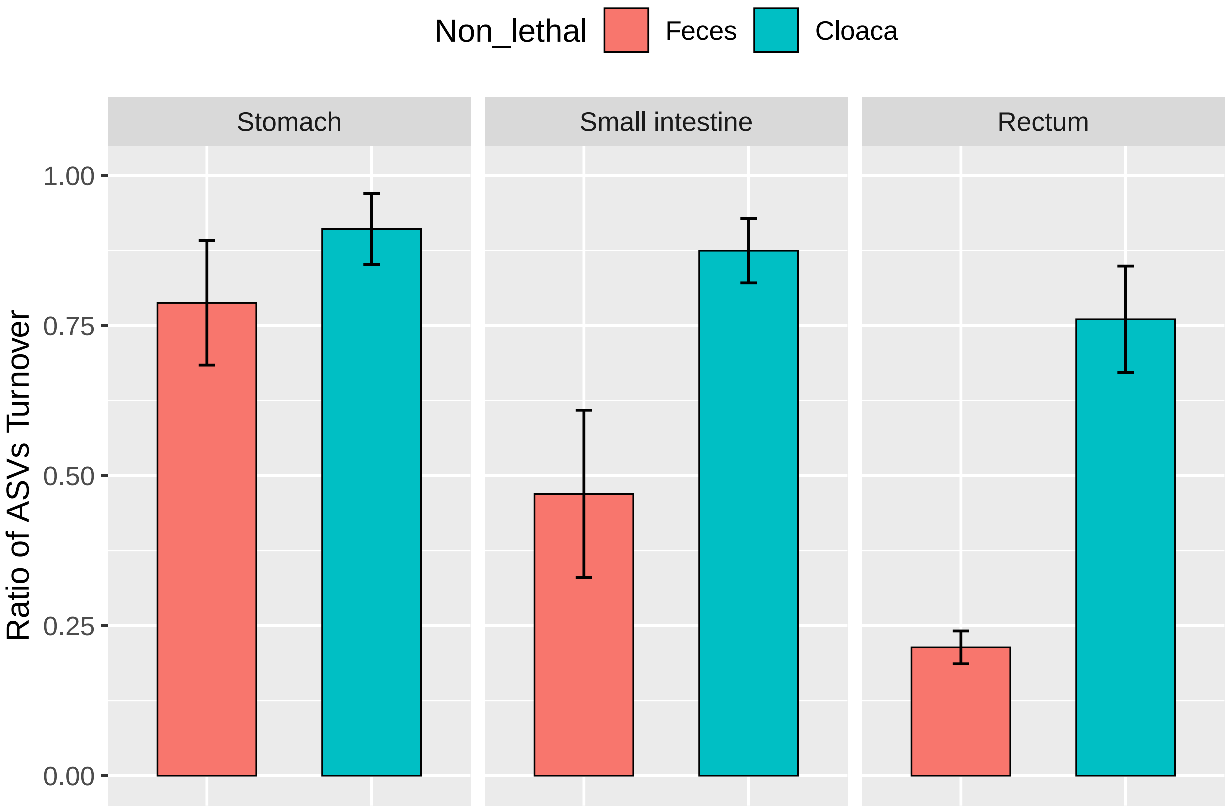
```
library(cowplot)
Reg <- plot_grid(Swab_Rectum, Swab_Small_intestine, Swab_Stomach,
                 Feces_Rectum, Feces_Small_intestine, Feces_Stomach)
print(Reg)
```



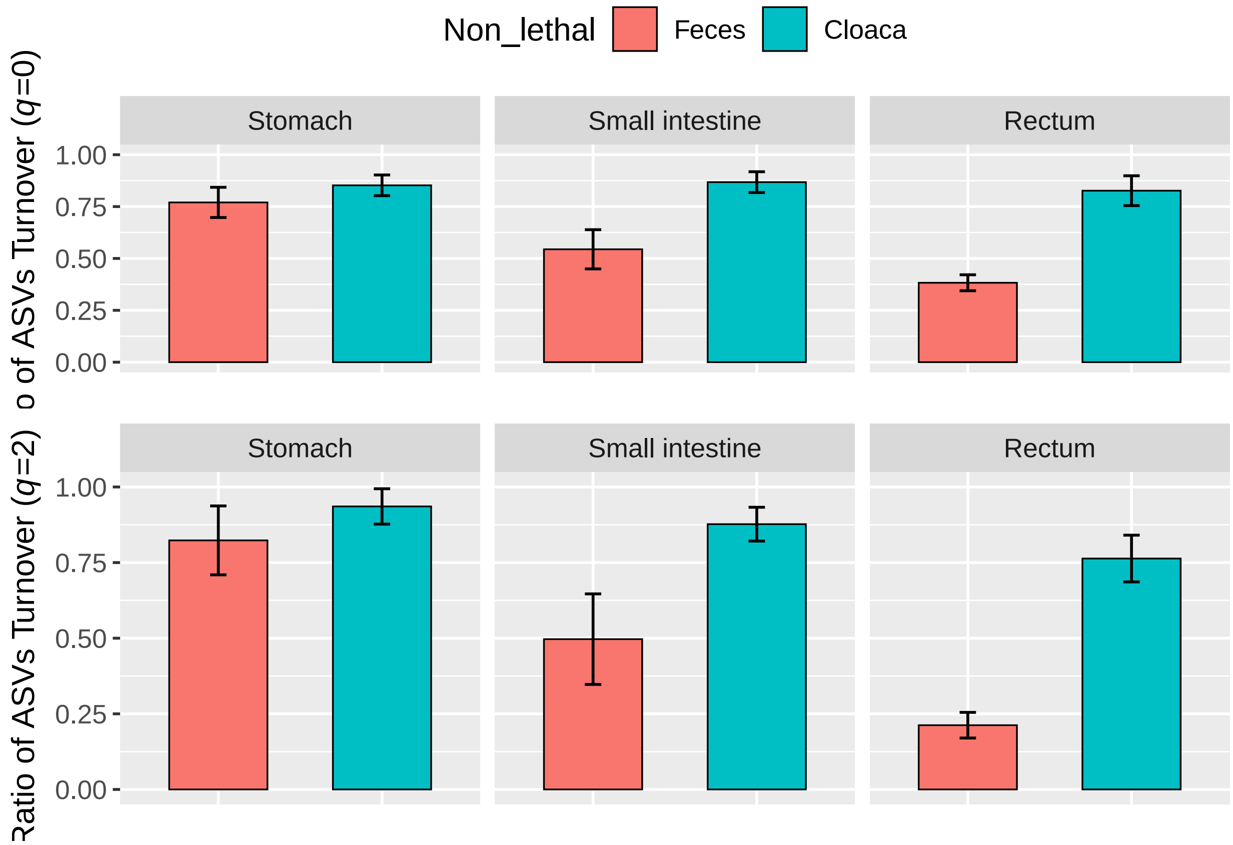
```
#ggsave("Reg.jpeg", width=20, height=18, dpi=300)
```

Turn Over

```
library(tidyverse)
library(ggpubr)

beta <- read.csv("../Data/INTER.csv", header = TRUE, check.names = F)

# Turnover al q=1 order
beta1=subset(beta, q==1)
beta1$DT<- factor(beta1$DT, levels = c("Stomach","Intestine", "Rectum"),
                  labels = c("Stomach", "Small intestine", "Rectum"))

#titulo1 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=1)"))

Turnover_q1 <- ggbarplot(subset(beta1, q==1), 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 = "Ratio of ASVs Turnover") +
  theme_gray() + theme(text = element_text (size = 12)) +
  theme(legend.position = "right",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  scale_y_continuous(limits = c(0,1))+
  geom_signif(test="wilcox.test") +theme(legend.position = "top")
Turnover_q1
```



```
beta0=subset(beta, q==0)
beta0$DT<- factor(beta0$DT, levels = c("Stomach","Intestine", "Rectum"),
                  labels = c("Stomach", "Small intestine", "Rectum"))

titulo0 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=0)"))

turnover0 <- ggbarplot(subset(beta0, q==0), 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 = 12)) +
  theme(legend.position = "right",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  scale_y_continuous(limits = c(0,1))+
  geom_signif(test="wilcox.test")+theme(legend.position = "top")

beta2=subset(beta, q==2)
beta2$DT<- factor(beta2$DT, levels = c("Stomach","Intestine", "Rectum"),
                  labels = c("Stomach", "Small intestine", "Rectum"))

titulo2 <- expression(paste("Ratio of ASVs Turnover (", italic("q"), "=2)"))

turnover2<- ggbarplot(subset(beta2, q==2), 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 = 12)) +
  theme(legend.position = "right",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())+
  scale_y_continuous(limits = c(0,1))+
  geom_signif(test="wilcox.test")+theme(legend.position = "none")


library(cowplot)
TurnoverFig_q02 <- plot_grid(turnover0,turnover2,
                         nrow = 2,ncol = 1)
TurnoverFig_q02
```



Venn Diagram

```
library(tidyverse)
# Loading files
# Core microbiota (50%)
swab_50 <- read.delim("../Data/core_otus_50swab.txt",
                      check.names = F, skip = 1) %>%rownames_to_column(
                        var = "ids")
feces_50 <- read.delim("../Data/core_otus_50_feces.txt",
                       check.names = F, skip = 1) %>%rownames_to_column(
                         var = "ids")
rectum_50 <- read.delim("../Data/core_otus_50_rectum.txt",
                        check.names = F, skip = 1) %>%rownames_to_column(
                          var = "ids")
smallint_50 <- read.delim("../Data/core_otus_50_smallintestine.txt",
                          check.names = F, skip = 1) %>%rownames_to_column(
                            var = "ids")
stomach_50 <- read.delim("../Data/core_otus_50_stomach.txt",
                         check.names = F, skip = 1) %>%rownames_to_column(
                           var = "ids")

# Create Venn Diagramm

library(VennDiagram)
```

```r
venn.plot_50 <- venn.diagram(
  x = list(Swab = swab_50$ids,
           Feces = feces_50$ids,
           Rectum = rectum_50$ids,
           Intestine = smallint_50$ids,
           Stomach = stomach_50$ids),
  category.names = c(
    expression(bold("Swab")),
    expression(bold("Feces")),
    expression(bold("Rectum")),
    expression(bold("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("yellow", "purple", "green", "black", "red"),
  cex = 1.5,
  fontface = "bold",
  fontfamily = "sans",
  cat.cex = 2,
  cat.fontface = "bold",
  cat.default.pos = "outer",
  cat.pos = c(-27, 27, 135, -125, -125),
  cat.dist = c(0.055, 0.055, 0.085, 0.060, 0.06),
  cat.fontfamily = "sans")
```

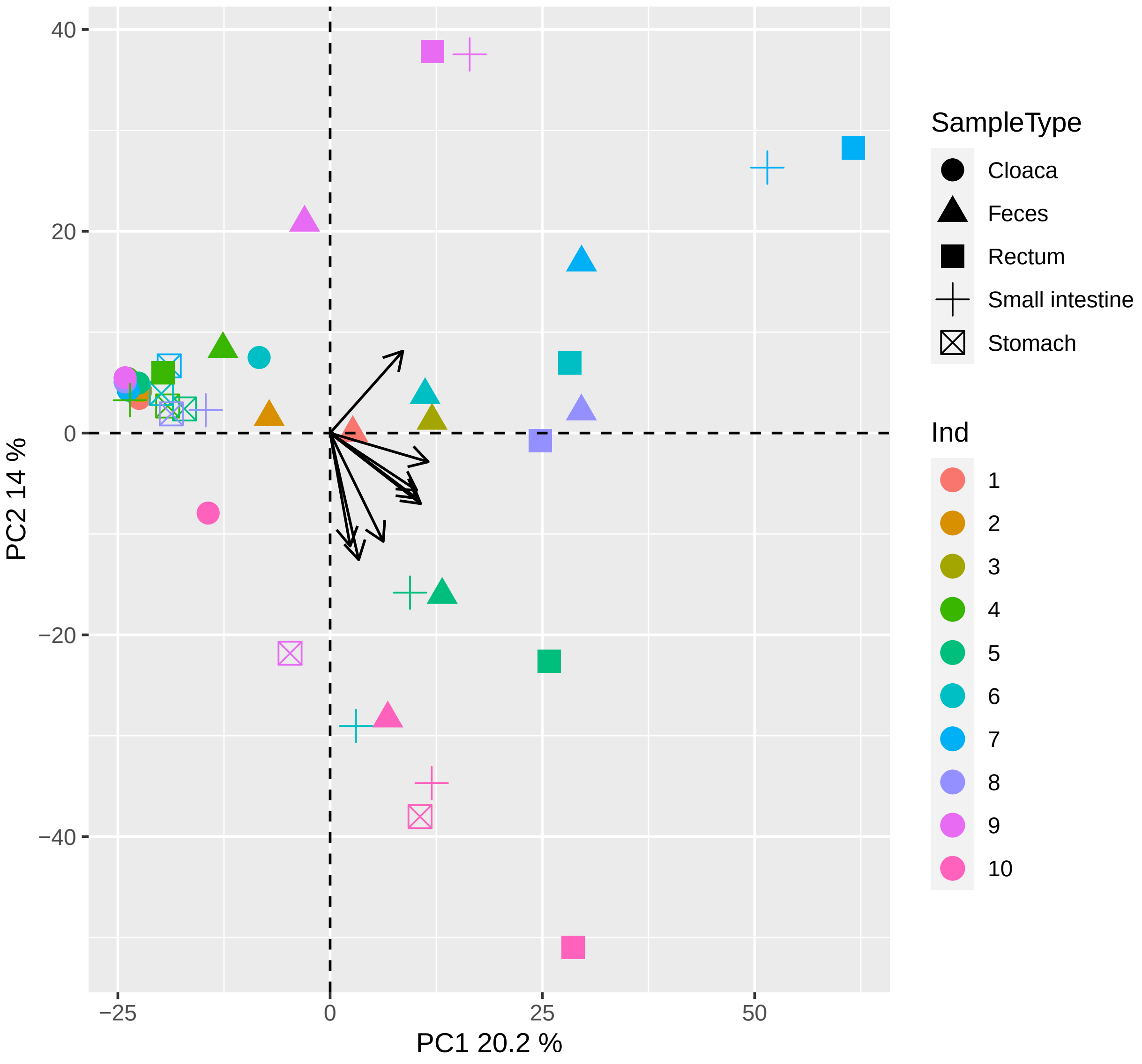
Beta Diversity

```r
## Loading libraries
library(tidyverse)
library(compositions)
library(zCompositions)
library(ALDEx2)
library(CoDaSeq)

# Loading files
otutable <- read.delim("../Data/otutable-taxonomy_ultima.txt",
                       row.names = 1) %>% dplyr::select(-taxonomy)
metadata <- read.csv("../Data/Metadatos1.csv", check.names = F)
metadata$Ind <- as.factor(metadata$Ind)

metadata<- metadata %>% mutate(SampleType=case_when(
  SampleType == "Swab"~"Cloaca",
  TRUE~as.character(SampleType)))
```

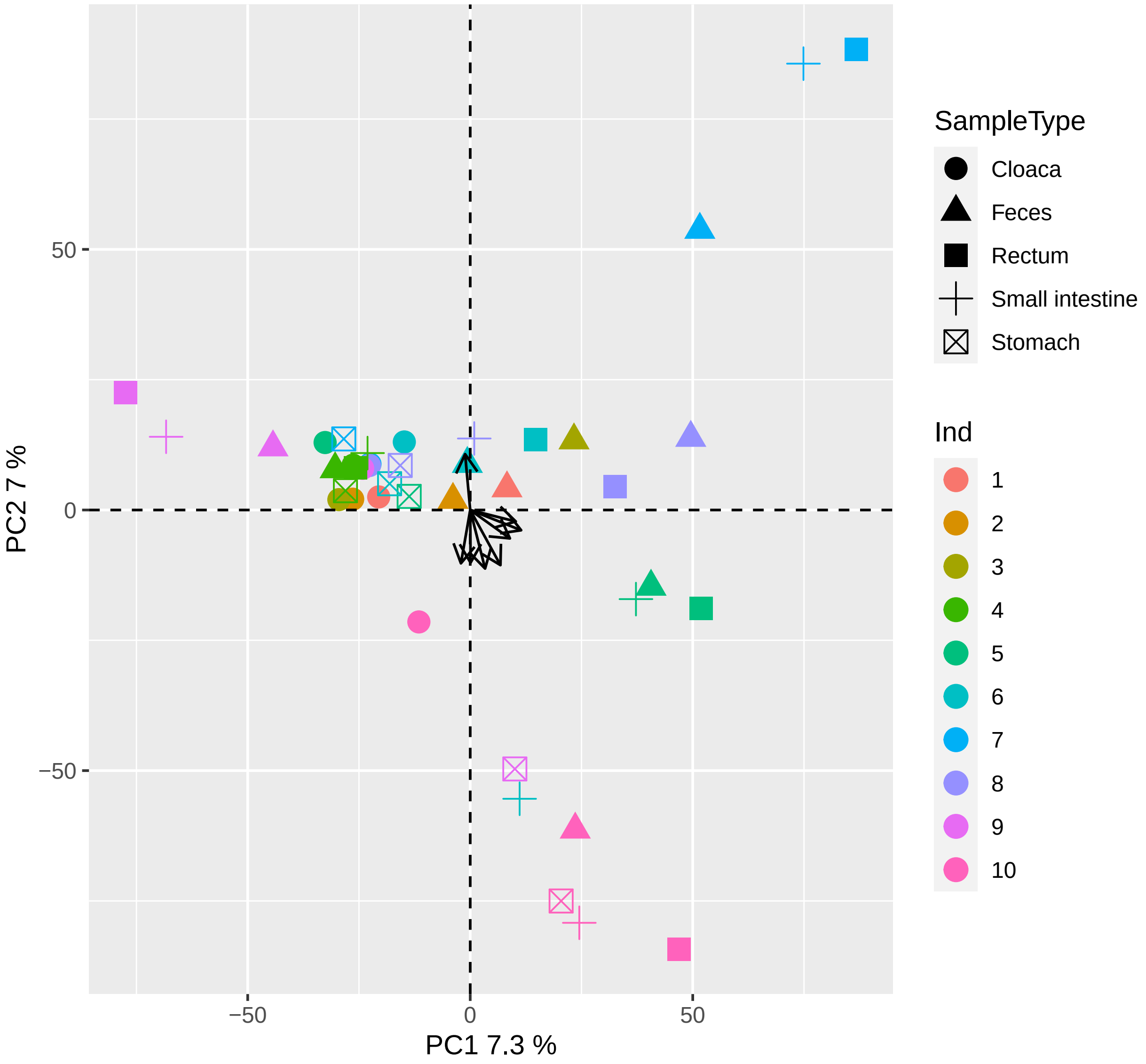
```r
metadata$Library <- as.factor(metadata$Library)
metadata$SampleType <- as.factor(metadata$SampleType)

taxonomy <- read.delim("../Data/taxonomy_ultima.txt", check.names = F) %>% unite(
  taxa, Kingdom:Species, remove = F, sep = ";")

# Write_tsv(metadata, "metadata.tsv")

taxonomy2 <- taxonomy %>%
  mutate_all(funs(str_replace(., "k__Bacteria;", "")))%>%
  mutate_all(funs(str_replace(., "p__", "")))%>%
  mutate_all(funs(str_replace(., "c__", "")))%>%
  mutate_all(funs(str_replace(., "o__", "")))%>%
  mutate_all(funs(str_replace(., "f__", "")))%>%
  mutate_all(funs(str_replace(., "g__", "")))%>%
  mutate_all(funs(str_replace(., "s__", "")))%>%
  mutate_all(funs(str_replace(., "; ; ;", "")))%>%
  mutate_all(funs(str_replace(., "; ; ", "")))

###### Transforming data "codaSeq.clr/compositional data" #####

d.pro <- cmultRepl(t(otutable), method = "CZM", output = "p-counts")
```

```
## No. corrected values:  771
```

```r
d.clr.abund.codaseq <- codaSeq.clr(x= d.pro, samples.by.row = F)

# Run a PCA with codaSeq.clr
pcx.abund <- prcomp(d.clr.abund.codaseq)

# Labels to PCA axis

PC1 <- paste("PC1", round(sum(
  pcx.abund$sdev[1] ^2) / mvar(d.clr.abund.codaseq) * 100, 1), "%")
PC2 <- paste("PC2", round(
  sum(pcx.abund$sdev[2] ^2) / mvar(d.clr.abund.codaseq) * 100, 1), "%")

# Create the base plot with only the arrows
pca_plot_codaSeq.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$x) %>% rownames_to_column(var = "SampleID") %>%
      left_join(metadata, by = "SampleID"),
    aes(x=PC1, y=PC2, shape=SampleType, color =Ind),
    size=4) +
  geom_vline(xintercept = 0, linetype = 2) +    #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  # ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
  # segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data=data.frame(pcx.abund$rotation) %>% rownames_to_column(
    var="#OTU ID") %>%
                   mutate(a=sqrt(PC1^2+PC2^2)) %>% # calculate the distance from the origin
                   top_n(8, a) %>% #keep 8 furthest away points
                   mutate(PC1=PC1*100, PC2=PC2*100) %>%
                   left_join(taxonomy2),
                 aes(x=0, xend=PC1, y=0, yend=PC2),
                 arrow = arrow(length = unit(0.3,"cm")))

print(pca_plot_codaSeq.clr)
```



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

#### 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))

# Run a PCA with codaSeq.clr
pcx.abund.aldex <- prcomp(aldex.clr.transform.data)

# Labels to PCA axis

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), "%")

# 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$x) %>% rownames_to_column(var = "SampleID") %>%
      left_join(metadata, by = "SampleID"),
    aes(x=PC1, y=PC2, shape=SampleType, color =Ind),
    size=4) +
  geom_vline(xintercept = 0, linetype = 2) +   #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  # ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
  # segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data=data.frame(pcx.abund.aldex$rotation) %>% rownames_to_column(
    var="#OTU ID") %>%
                 mutate(a=sqrt(PC1^2+PC2^2)) %>% # calculate the distance from the origin
                 top_n(8, a) %>% #keep 8 furthest away points
                 mutate(PC1=PC1*100, PC2=PC2*100) %>%
                 left_join(taxonomy2),
               aes(x=0, xend=PC1, y=0, yend=PC2),
               arrow = arrow(length = unit(0.3,"cm")))

print(pca_plot_aldex.clr)
```

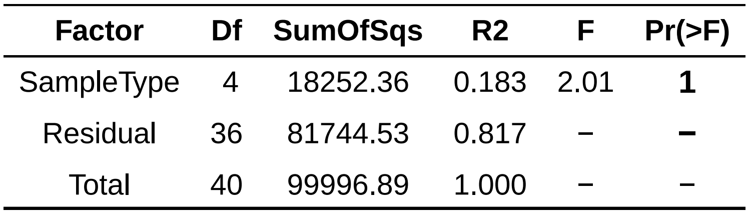


```
### PERMANOVA
set.seed(123)
library(vegan)
library(qiime2R)
meta_just<- data.frame(d.clr.abund.codaseq, check.names = F) %>% rownames_to_column(
  var = "SampleID") %>% inner_join(metadata) %>% rename(SampleID="SampleID" )
  library(vegan)
library(RVAideMemoire)
library(ggpubr)
perm<- how(nperm = 999)

setBlocks(perm)<- with(meta_just,SampleID)
permanova_ma<-adonis2(d.clr.abund.codaseq~SampleType,
                      data = meta_just, method = "euclidian",
                      permutations =perm) %>% round(., digits = 3) %>%replace(is.na(.), "-")
pairwsie<-RVAideMemoire::pairwise.perm.manova(dist(
  d.clr.abund.codaseq, method="euclidian"), meta_just$SampleType, p.method =  "BH", nperm = 999)
```

```r
pairwsie
```

```
##
##  Pairwise comparisons using permutation MANOVAs on a distance matrix
##
## data:  dist(d.clr.abund.codaseq, method = "euclidian") by meta_just$SampleType
## 999 permutations
##
##                 Cloaca Feces  Rectum Small intestine
## Feces           0.0025 -      -      -
## Rectum          0.0025 0.7322 -      -
## Small intestine 0.0025 0.5750 0.7870 -
## Stomach         0.0025 0.0080 0.0150 0.3957
##
## P value adjustment method: BH
```

```r
Permanova_table<-data.frame(permanova_ma, check.names = F) %>% rownames_to_column(
  var="Factor") %>% ggtexttable(., rows = NULL, theme = ttheme("blank")) %>%
  tab_add_hline(at.row = 1:2, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 6, face = "bold") %>%
  table_cell_font(., row = 2, column = 6, face = "bold") %>%
  tab_add_hline(at.row = c(4), row.side = "bottom", linewidth = 3, linetype = 1)

Permanova_table
```



Factor	Df	SumOfSqs	R2	F	Pr(>F)
SampleType	4	18252.36	0.183	2.01	**1**
Residual	36	81744.53	0.817	-	**-**
Total	40	99996.89	1.000	-	-

```r
Pairwsie_permanova<- data.frame(pairwsie$p.value, check.names = F)%>% round(
  .,digits = 3) %>%replace(
  is.na(.), "-")  %>% rownames_to_column(
    var="Type") %>% ggtexttable(., rows = NULL, theme = ttheme("blank")) %>%
  tab_add_hline(at.row = 1:2, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 4, column = 2, face = "bold") %>%
  table_cell_font(., row = 5, column = 2, face = "bold") %>%
  table_cell_font(., row = 5, column = 3, face = "bold") %>%
  table_cell_font(., row = 4, column = 3, face = "bold") %>%
  table_cell_font(., row = 5, column = 5, face = "bold") %>%
  table_cell_font(., row = 5, column = 4, face = "bold")  %>%
  table_cell_font(., row = 2:5, column = 1, face = "bold")  %>%
  tab_add_hline(at.row = c(5), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(
    text = "*p values in Bold are significant using \n an alpha value of 0.05",
    size = 10, face = "italic")
Pairwsie_permanova
```

Type	Cloaca	Feces	Rectum	Small intestine
Feces	0.002	–	–	–
Rectum	0.002	0.732	–	–
Small intestine	**0.002**	**0.575**	0.787	–
Stomach	**0.002**	**0.008**	**0.015**	**0.396**

**p values in Bold are significant using an alpha value of 0.05*

Alpha Diversity

```
## Loading libraries

library(tidyverse)
library(ggpubr)

#loading files

alpha_div <- read.csv("../Data/Hill_numbers_q012.csv", header = TRUE, check.names = F)
alpha <- read.csv("../Data/Hill_numbers_q012.csv") %>% dplyr::select(SampleID, q0, q1, q2)
alpha_div$SampleType <- as.factor(alpha_div$SampleType)
metadata <- read.csv("../Data/Metadatos1.csv",check.names = F)
alpha <- alpha %>% inner_join(metadata, by = c("SampleID"="SampleID"))

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

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q0
## W = 0.91958, p-value = 0.006614
```

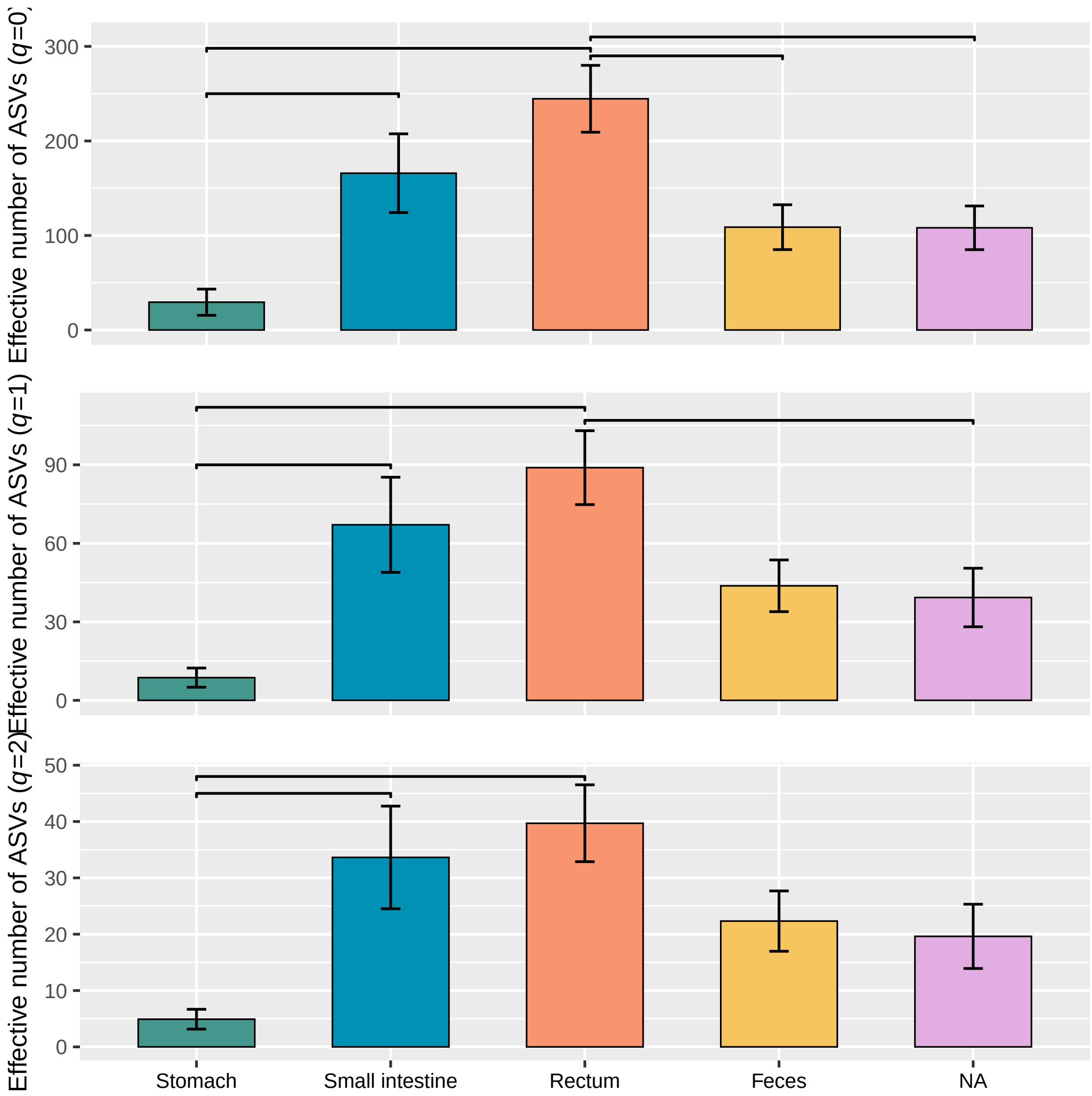
```
shapiro.test(x =alpha$q1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q1
## W = 0.87443, p-value = 0.000317
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha$q2
## W = 0.88502, p-value = 0.000617
```

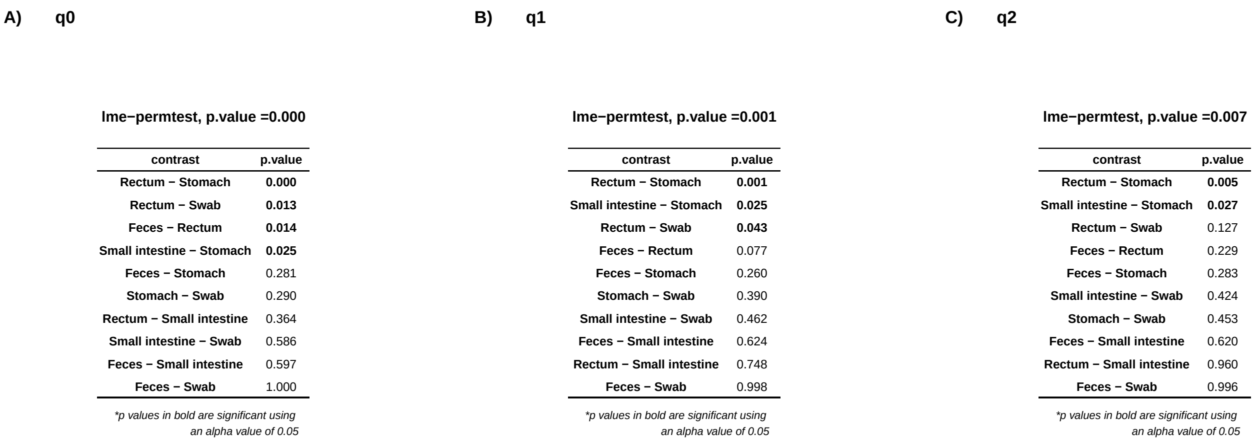
```
titulo <- expression(paste("Effective number of ASVs (", italic("q"), "=0)"))

HillNumb_q0 <- ggbarplot(alpha, x= "SampleType", y= "q0",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Swab"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = titulo) +
  theme_gray() + theme(text = element_text (size = 10)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  geom_signif(annotations=c("","", "", ""), tip_length = 0.01, vjust = 0.9,
              y_position=c(298, 310, 250, 290),
              xmin=c(1, 3, 1, 3), xmax=c(3, 5, 2, 4))

titulo1 <- expression(paste("Effective number of ASVs (", italic("q"), "=1)"))
HillNumb_q1 <- ggbarplot(alpha, x= "SampleType", y= "q1",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Swab"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = titulo1) +
  theme_gray() + theme(text = element_text (size = 10)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  geom_signif(annotations=c("","",""), tip_length = 0.01, vjust = 0.9,
              y_position=c(112, 90, 107),
              xmin=c(1,1,3), xmax=c(3,2,5))

titulo2 <- expression(paste("Effective number of ASVs (", italic("q"), "=2)"))
HillNumb_q2 <- ggbarplot(alpha, x= "SampleType", y= "q2",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c("Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = titulo2) +
  theme_gray() + theme(text = element_text (size = 10))+
  geom_signif(annotations=c(""), tip_length = 0.01, vjust = 0.2,
              y_position=c(48, 45),
              xmin=c(1,1), xmax=c(3,2))+ theme(axis.text.x = element_text(color = "black"))
#theme(legend.position = "none",
   #     axis.ticks.x = element_blank(),
  #      axis.text.x = element_blank())

library(cowplot)
Graphics_boxplot <- plot_grid(HillNumb_q0, HillNumb_q1, HillNumb_q2,
                              nrow = 3, ncol = 1,
                              label_size = 10, rel_heights = c(1, 1, 1))
print(Graphics_boxplot)
```



```
# Linear mixed model approach
library(lme4)
library(nlme)
library(cowplot)
library(pgirmess) # includes PermTest()
library(emmeans)

q0_lme <- lme(q0~ SampleType, random = ~1 |Ind, data = alpha)
q0_lme_perm <- PermTest(q0_lme)
q0_lme_means <- emmeans(q0_lme, pairwise ~ SampleType)

q1_lme<- lme(q1~ SampleType, random=~1 |Ind, data = alpha)
q1_lme_perm <- PermTest(q1_lme)
q1_lme_means <- emmeans(q1_lme, pairwise ~ SampleType)

q2_lme <- lme(q2~ SampleType, random=~1 |Ind, data = alpha)
q2_lme_perm <- PermTest(q2_lme)
```

```r
q2_lme_means <- emmeans(q2_lme, pairwise ~ SampleType)
```

```r
# Tables summarizing results

library(kableExtra)
install.packages("kableExtra")

q0_lme_means.t <- data.frame(q0_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(
  var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(
    var="contrast") %>% ggtexttable(
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q0_lme_perm$resultats$p.value[2],
                                     digits=3, nsmall=3)) , face = "bold",
    padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(text = "*p values in bold are significant using \n an alpha value of 0.05",
                   size = 9, face = "italic")



q1_lme_means.t <- data.frame(q1_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(
  var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(
    var="contrast") %>% ggtexttable(
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q1_lme_perm$resultats$p.value[2],
                                     digits=3, nsmall=3)) , face = "bold",
    padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 8, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(text = "*p values in bold are significant using \n an alpha value of 0.05",
                   size = 9, face = "italic")


q2_lme_means.t <- data.frame(q2_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(
  var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(
    var="contrast") %>% ggtexttable(
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q2_lme_perm$resultats$p.value[2],
                                  digits=3, nsmall=3)) , face = "bold",
    padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 8, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "plain", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(text = "*p values in bold are significant using \n an alpha value of 0.05",
                 size = 9, face = "italic")


library(cowplot)

comparisons <- plot_grid(q0_lme_means.t,q1_lme_means.t,q2_lme_means.t,
                  nrow = 2,ncol = 3, labels =
                   c("A)       q0",
                     "B)       q1",
                     "C)       q2"),
                  rel_heights = c(1,1.7))
print(comparisons)
```



A) q0

lme−permtest, p.value =0.000

contrast	p.value
Rectum − Stomach	**0.000**
Rectum − Swab	**0.013**
Feces − Rectum	**0.014**
Small intestine − Stomach	**0.025**
Feces − Stomach	0.281
Stomach − Swab	0.290
Rectum − Small intestine	0.364
Small intestine − Swab	0.586
Feces − Small intestine	0.597
Feces − Swab	1.000

**p values in bold are significant using an alpha value of 0.05*

B) q1

lme−permtest, p.value =0.001

contrast	p.value
Rectum − Stomach	**0.001**
Small intestine − Stomach	**0.025**
Rectum − Swab	**0.043**
Feces − Rectum	0.077
Feces − Stomach	0.260
Stomach − Swab	0.390
Small intestine − Swab	0.462
Feces − Small intestine	0.624
Rectum − Small intestine	0.748
Feces − Swab	0.998

**p values in bold are significant using an alpha value of 0.05*

C) q2

lme−permtest, p.value =0.007

contrast	p.value
Rectum − Stomach	**0.005**
Small intestine − Stomach	**0.027**
Rectum − Swab	0.127
Feces − Rectum	0.229
Feces − Stomach	0.283
Small intestine − Swab	0.424
Stomach − Swab	0.453
Feces − Small intestine	0.620
Rectum − Small intestine	0.960
Feces − Swab	0.996

**p values in bold are significant using an alpha value of 0.05*

Functional diversity

```
#Functional diversity calculation
Picrust <- read.delim("../Data/EC_predicted.tsv", check.names = F, row.names = 1)
totutable <- read.delim("otutable-taxonomy_ultima.txt", check.names = F) %>% dplyr::select(
  -taxonomy) %>% column_to_rownames(var = "#OTU ID") %>% t()
```

```
totutable <- totutable[ , match(rownames(Picrust), colnames(totutable))]
metadata<- read.csv("Metadatos1.csv", check.names = F) %>% mutate(
  SampleType=case_when(
    SampleType=="Swab"~"Cloaca",
    TRUE~as.character(SampleType)))
alpha <- alpha %>% rownames_to_column(var="SampleID") %>% inner_join(metadata, by = c("SampleID"="Sampl

func_q0 <- hill_func(totutable, traits = Picrust, q = 0)
func_q1 <- hill_func(totutable, traits = Picrust, q = 1)
func_q2 <- hill_func(totutable, traits = Picrust, q = 2)

funq0<- func_q0 %>% t() %>% as.data.frame() %>% dplyr::select(
  q0=MD_q) %>% rownames_to_column(var = "SampleID")
funq1<- func_q1 %>% t() %>% as.data.frame() %>% dplyr::select(
  q1=MD_q) %>% rownames_to_column(var = "SampleID")
funq2<- func_q2 %>% t() %>% as.data.frame() %>% dplyr::select(
  q2=MD_q) %>% rownames_to_column(var = "SampleID")

functional_div<- funq0 %>% inner_join(funq1) %>% inner_join(funq2) %>% inner_join(metadata)
```

```
# Loading files
library(tidyverse)
library(ggpubr)

alpha <- read.csv("../Data/Functional_div.csv", header = TRUE, check.names = F)

library(lme4)
library(nlme)
library(cowplot)
library(pgirmess) # includes PermTest()
library(emmeans)

q0_lme <- lme(q0~ SampleType, random = ~1 |Ind, data = alpha)
summary(q0_lme)
```

```
## Linear mixed-effects model fit by REML
##   Data: alpha
##        AIC      BIC    logLik
##   728.0236 739.1082 -357.0118
##
## Random effects:
##  Formula: ~1 | Ind
##         (Intercept) Residual
## StdDev:    2005.175 3884.036
##
## Fixed effects:  q0 ~ SampleType
##                                 Value Std.Error DF   t-value p-value
## (Intercept)                 11256.601  1382.261 27  8.143613  0.0000
## SampleTypeRectum             3866.005  1943.214 27  1.989490  0.0569
## SampleTypeSmall intestine   -1513.960  1943.214 27 -0.779101  0.4427
## SampleTypeStomach           -6482.819  1943.214 27 -3.336132  0.0025
## SampleTypeSwab              -9512.351  1736.994 27 -5.476330  0.0000
##  Correlation:
```

```
##                            (Intr) SmplTR SmpTSi SmplTypSt
## SampleTypeRectum           -0.562                        
## SampleTypeSmall intestine -0.562  0.429                 
## SampleTypeStomach          -0.562  0.429  0.429          
## SampleTypeSwab             -0.628  0.447  0.447  0.447   
## 
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max 
## -2.13189542 -0.31668486 -0.08895812  0.42879065  2.14569568 
## 
## Number of Observations: 41
## Number of Groups: 10
```

```
q0_lme_perm <- PermTest(q0_lme)
q0_lme_means <- emmeans(q0_lme, pairwise ~ SampleType)

q1_lme<- lme(q1~ SampleType, random=~1 |Ind, data = alpha)
summary(q1_lme)
```

```
## Linear mixed-effects model fit by REML
##   Data: alpha 
##        AIC      BIC    logLik
##   663.5015 674.5861 -324.7508
## 
## Random effects:
##  Formula: ~1 | Ind
##         (Intercept) Residual
## StdDev:    1016.576  1530.66
## 
## Fixed effects:  q1 ~ SampleType 
##                               Value Std.Error DF   t-value p-value
## (Intercept)                4663.274  581.0634 27  8.025414  0.0000
## SampleTypeRectum           1139.501  769.7570 27  1.480339  0.1504
## SampleTypeSmall intestine  -241.641  769.7570 27 -0.313918  0.7560
## SampleTypeStomach         -2695.509  769.7570 27 -3.501766  0.0016
## SampleTypeSwab            -4026.757  684.5318 27 -5.882498  0.0000
##  Correlation: 
##                            (Intr) SmplTR SmpTSi SmplTypSt
## SampleTypeRectum           -0.524                        
## SampleTypeSmall intestine -0.524  0.435                 
## SampleTypeStomach          -0.524  0.435  0.435          
## SampleTypeSwab             -0.589  0.445  0.445  0.445   
## 
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max 
## -2.1461456 -0.3045242 -0.1020280  0.5267705  2.2200294 
## 
## Number of Observations: 41
## Number of Groups: 10
```

```
q1_lme_perm <- PermTest(q1_lme)
q1_lme_means <- emmeans(q1_lme, pairwise ~ SampleType)
```

```
q2_lme <- lme(q2~ SampleType, random=~1 |Ind, data = alpha)
summary(q2_lme)
```

```
## Linear mixed-effects model fit by REML
##   Data: alpha
##        AIC      BIC    logLik
##   616.7461 627.8308 -301.3731
##
## Random effects:
##  Formula: ~1 | Ind
##         (Intercept) Residual
## StdDev:    574.1851 787.8532
##
## Fixed effects:  q2 ~ SampleType
##                               Value Std.Error DF   t-value p-value
## (Intercept)                2593.2093  308.2858 27  8.411706  0.0000
## SampleTypeRectum            293.9972  396.9835 27  0.740578  0.4653
## SampleTypeSmall intestine  -185.3674  396.9835 27 -0.466940  0.6443
## SampleTypeStomach         -1425.9380  396.9835 27 -3.591933  0.0013
## SampleTypeSwab            -2190.7561  352.3387 27 -6.217757  0.0000
##  Correlation:
##                           (Intr) SmplTR SmpTSi SmplTypSt
## SampleTypeRectum          -0.507
## SampleTypeSmall intestine -0.507  0.437
## SampleTypeStomach         -0.507  0.437  0.437
## SampleTypeSwab            -0.571  0.444  0.444  0.444
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.92399455 -0.50381036 -0.05937454  0.54451412  2.15113054
##
## Number of Observations: 41
## Number of Groups: 10
```

```
q2_lme_perm <- PermTest(q2_lme)
q2_lme_means <- emmeans(q2_lme, pairwise ~ SampleType)

q0_lme_perm;q1_lme_perm;q2_lme_perm
```

```
##
## Monte-Carlo test
##
## Call:
## PermTest.lme(obj = q0_lme)
##
## Based on 1000 replicates
## Simulated p-value:
##             p.value
## (Intercept)       0
## SampleType        0
```

```
##
```

```
## Monte-Carlo test
##
## Call:
## PermTest.lme(obj = q1_lme)
##
## Based on 1000 replicates
## Simulated p-value:
##             p.value
## (Intercept)   0.001
## SampleType    0.000
```

```
##
## Monte-Carlo test
##
## Call:
## PermTest.lme(obj = q2_lme)
##
## Based on 1000 replicates
## Simulated p-value:
##             p.value
## (Intercept)   0.001
## SampleType    0.000
```

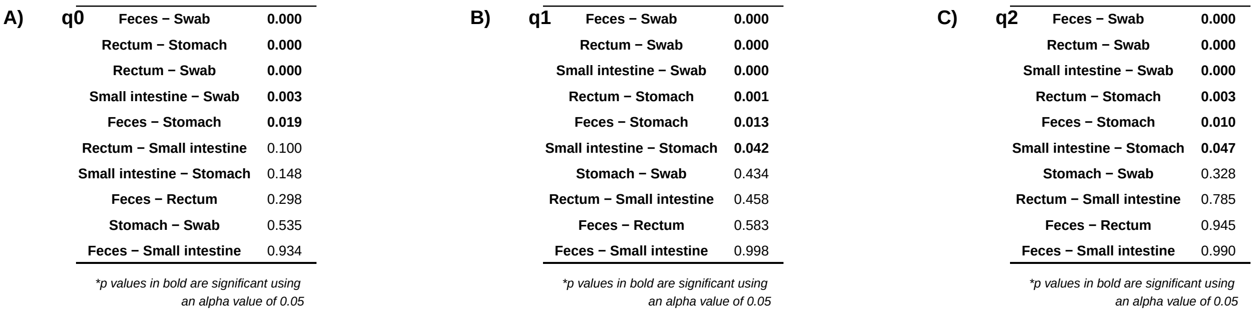
```
q2_lme_means; q1_lme_means;q2_lme_means
```

```
## $emmeans
##  SampleType      emmean  SE df lower.CL upper.CL
##  Feces             2593 308  9     1896     3291
##  Rectum            2887 358  9     2076     3698
##  Small intestine   2408 358  9     1597     3219
##  Stomach           1167 358  9      356     1978
##  Swab               402 308  9     -295     1100
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast                  estimate  SE df t.ratio p.value
##  Feces - Rectum                -294 397 27  -0.741  0.9449
##  Feces - Small intestine        185 397 27   0.467  0.9897
##  Feces - Stomach               1426 397 27   3.592  0.0104
##  Feces - Swab                  2191 352 27   6.218  <.0001
##  Rectum - Small intestine       479 421 27   1.138  0.7850
##  Rectum - Stomach              1720 421 27   4.084  0.0030
##  Rectum - Swab                 2485 397 27   6.259  <.0001
##  Small intestine - Stomach     1241 421 27   2.946  0.0473
##  Small intestine - Swab        2005 397 27   5.052  0.0002
##  Stomach - Swab                 765 397 27   1.927  0.3281
##
## Degrees-of-freedom method: containment
## P value adjustment: tukey method for comparing a family of 5 estimates
```

```
## $emmeans
##  SampleType      emmean  SE df lower.CL upper.CL
##  Feces             4663 581  9     3349     5978
##  Rectum            5803 679  9     4266     7340
##  Small intestine   4422 679  9     2885     5959
##  Stomach           1968 679  9      431     3505
##  Swab               637 581  9     -678     1951
## 
## Degrees-of-freedom method: containment 
## Confidence level used: 0.95 
## 
## $contrasts
##  contrast                  estimate  SE df t.ratio p.value
##  Feces - Rectum               -1140 770 27  -1.480  0.5834
##  Feces - Small intestine        242 770 27   0.314  0.9978
##  Feces - Stomach               2696 770 27   3.502  0.0129
##  Feces - Swab                  4027 685 27   5.882  <.0001
##  Rectum - Small intestine      1381 818 27   1.688  0.4577
##  Rectum - Stomach              3835 818 27   4.687  0.0006
##  Rectum - Swab                 5166 770 27   6.712  <.0001
##  Small intestine - Stomach     2454 818 27   2.999  0.0420
##  Small intestine - Swab        3785 770 27   4.917  0.0003
##  Stomach - Swab                1331 770 27   1.729  0.4337
## 
## Degrees-of-freedom method: containment 
## P value adjustment: tukey method for comparing a family of 5 estimates
```

```
## $emmeans
##  SampleType      emmean  SE df lower.CL upper.CL
##  Feces             2593 308  9     1896     3291
##  Rectum            2887 358  9     2076     3698
##  Small intestine   2408 358  9     1597     3219
##  Stomach           1167 358  9      356     1978
##  Swab               402 308  9     -295     1100
## 
## Degrees-of-freedom method: containment 
## Confidence level used: 0.95 
## 
## $contrasts
##  contrast                  estimate  SE df t.ratio p.value
##  Feces - Rectum                -294 397 27  -0.741  0.9449
##  Feces - Small intestine        185 397 27   0.467  0.9897
##  Feces - Stomach               1426 397 27   3.592  0.0104
##  Feces - Swab                  2191 352 27   6.218  <.0001
##  Rectum - Small intestine       479 421 27   1.138  0.7850
##  Rectum - Stomach              1720 421 27   4.084  0.0030
##  Rectum - Swab                 2485 397 27   6.259  <.0001
##  Small intestine - Stomach     1241 421 27   2.946  0.0473
##  Small intestine - Swab        2005 397 27   5.052  0.0002
##  Stomach - Swab                 765 397 27   1.927  0.3281
## 
## Degrees-of-freedom method: containment 
## P value adjustment: tukey method for comparing a family of 5 estimates
```

```
library(kableExtra)

q0_lme_means.t <- data.frame(
  q0_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(var="contrast"
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q0_lme_perm$resultats$p.value[2], digits=3, nsmall=3)) ,
    face = "bold", padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(
    text = "*p values in bold are significant using \n an alpha value of 0.05",
                   size = 9, face = "italic")

q1_lme_means.t <- data.frame(
  q1_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(
    var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(
    var="contrast") %>% ggtexttable(
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q1_lme_perm$resultats$p.value[2], digits=3, nsmall=3)) ,
    face = "bold", padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 8, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(
    text = "*p values in bold are significant using \n an alpha value of 0.05",
                   size = 9, face = "italic")


q2_lme_means.t <- data.frame(
  q2_lme_means$contrasts)[,c(1,6)] %>% column_to_rownames(
    var = "contrast") %>% round(
  .,digits = 3) %>%replace(  is.na(.), "-")  %>% arrange(p.value) %>% rownames_to_column(
    var="contrast") %>% ggtexttable(
    ., rows = NULL, theme = ttheme("blank", base_size = 10)) %>%
  tab_add_title(text = paste0(
    "lme-permtest, p.value =",format(q2_lme_perm$resultats$p.value[2], digits=3, nsmall=3)) ,
    face = "bold", padding = unit(3, "line")) %>%
  tab_add_hline(at.row = 2:3, row.side = "top", linewidth = 2)%>%
  table_cell_font(., row = 3, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 4, column = 2, face = "bold",size = 10) %>%
  table_cell_font(., row = 7, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 8, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 5, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 6, column = 2, face = "bold", size = 10) %>%
  table_cell_font(., row = 3:tab_nrow(.), column = 1, face = "bold", size = 10)  %>%
  tab_add_hline(at.row = c(12), row.side = "bottom", linewidth = 3, linetype = 1) %>%
  tab_add_footnote(
    text = "*p values in bold are significant using \n an alpha value of 0.05",
                   size = 9, face = "italic")
library(cowplot)

Comparisons_Funct_Div <- plot_grid(q0_lme_means.t,q1_lme_means.t,q2_lme_means.t,
                         nrow = 2,ncol = 3, labels =
                           c("A)       q0",
                             "B)       q1",
                             "C)       q2"),
                         rel_heights = c(1,1.7))
print(Comparisons_Funct_Div)
```



A) **q0**

Feces − Swab	**0.000**
Rectum − Stomach	**0.000**
Rectum − Swab	**0.000**
Small intestine − Swab	**0.003**
Feces − Stomach	**0.019**
Rectum − Small intestine	0.100
Small intestine − Stomach	0.148
Feces − Rectum	0.298
Stomach − Swab	0.535
Feces − Small intestine	0.934

**p values in bold are significant using an alpha value of 0.05*

B) **q1**

Feces − Swab	**0.000**
Rectum − Swab	**0.000**
Small intestine − Swab	**0.000**
Rectum − Stomach	**0.001**
Feces − Stomach	**0.013**
Small intestine − Stomach	**0.042**
Stomach − Swab	0.434
Rectum − Small intestine	0.458
Feces − Rectum	0.583
Feces − Small intestine	0.998

**p values in bold are significant using an alpha value of 0.05*

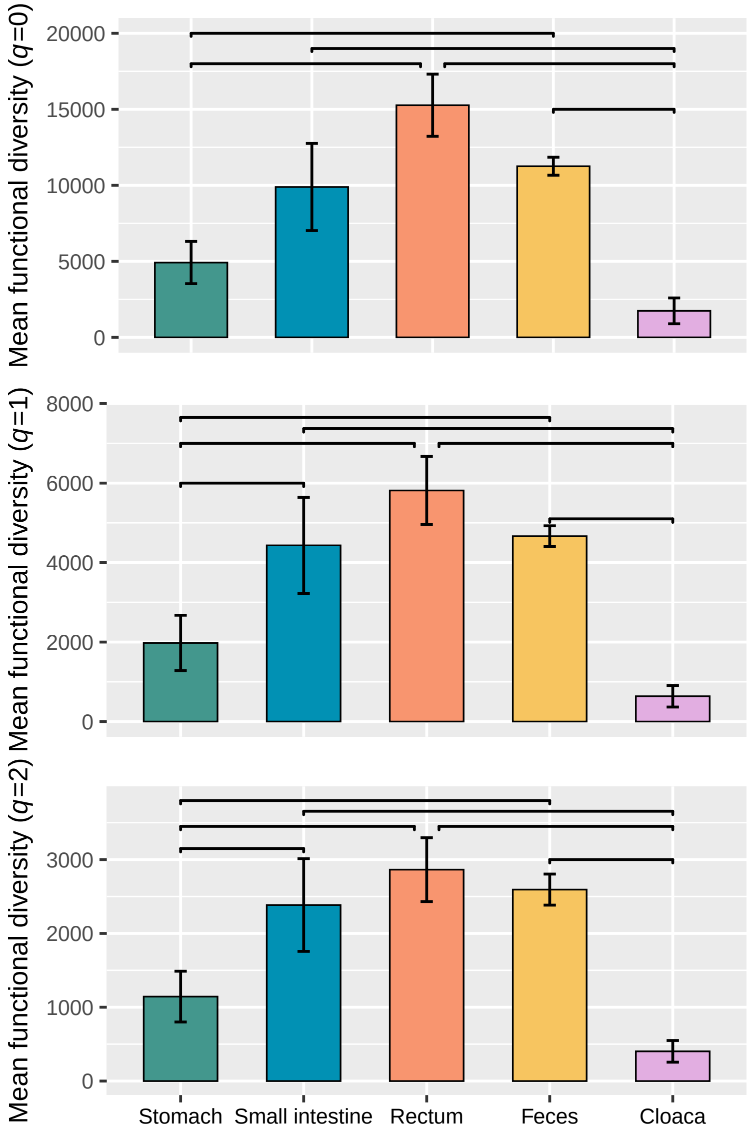
C) **q2**

Feces − Swab	**0.000**
Rectum − Swab	**0.000**
Small intestine − Swab	**0.000**
Rectum − Stomach	**0.003**
Feces − Stomach	**0.010**
Small intestine − Stomach	**0.047**
Stomach − Swab	0.328
Rectum − Small intestine	0.785
Feces − Rectum	0.945
Feces − Small intestine	0.990

**p values in bold are significant using an alpha value of 0.05*

```
alpha <- read.csv("../Data/Functional_div.csv", header = TRUE, check.names = F)
alpha<- alpha %>% mutate(
  SampleType=case_when(
    SampleType=="Swab"~"Cloaca",
    TRUE~as.character(SampleType)))
tituloA <- expression(paste("Mean functional diversity (", italic("q"), "=0)"))
HillNumb_q0 <- ggbarplot(alpha, x= "SampleType", y= "q0",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c(
                           "Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = tituloA) +
  theme_gray() + theme(text = element_text (size = 10)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  geom_signif(annotations=c("","","","",""), tip_length = 0.01, vjust = 0.9,
              y_position=c(15000, 18000, 18000, 19000, 20000),
              xmin=c(4,3.1,1,2,1), xmax=c(5,5,2.9,5,4))
```

```r
tituloB <- expression(paste("Mean functional diversity (", italic("q"), "=1)"))
HillNumb_q1 <- ggbarplot(alpha, x= "SampleType", y= "q1",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c(
                           "Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = tituloB) +
  theme_gray() + theme(text = element_text (size = 10)) +
  theme(legend.position = "none",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank()) +
  geom_signif(annotations=c("","","","","",""), tip_length = 0.01, vjust = 0.9,
              y_position=c(5100,7000,7370,7000,7650,6000),
              xmin=c(4,3.1,2,1,1,1), xmax=c(5,5,5,2.9,4,2))

tituloC <- expression(paste("Mean functional diversity (", italic("q"), "=2)"))
HillNumb_q2 <- ggbarplot(alpha, x= "SampleType", y= "q2",
                         color = "black",  width = 0.6, lwd=0.3,
                         order = c(
                           "Stomach", "Small intestine", "Rectum", "Feces", "Cloaca"),
                         fill = c("#43978D","#0191B4","#F8956F", "#F7C560", "#E2AEE1"),
                         add = "mean_se") +
  labs(x = element_blank(), y = tituloC) +
  theme_gray() + theme(text = element_text (size = 10)) +
  geom_signif(annotations=c("","","","","",""), tip_length = 0.01, vjust = 0.2,
              y_position=c(3000,3450,3655,3450,3800,3150),
              xmin=c(4,3.1,2,1,1,1), xmax=c(5,5,5,2.9,4,2)) +
  theme(axis.text.x = element_text(color = "black"))

Boxplot_funct_div <- plot_grid(HillNumb_q0, HillNumb_q1, HillNumb_q2,
                                nrow = 3, ncol = 1,
                                label_size = 10, rel_heights = c(1, 1, 1))
print(Boxplot_funct_div)
```


Mean functional diversity (q=0)
20000
15000
10000
5000
0
Mean functional diversity (q=1)
8000
6000
4000
2000
0
Mean functional diversity (q=2)
3000
2000
1000
0
Stomach
Small intestine
Rectum
Feces
Cloaca