Supplemental File of

MicrobiotaProcess: A tidy framework for microbiome or other related ecology data analysis

Shuangbin Xu, Li Zhan, Wenli Tang, Zehan Dai, Lang Zhou, Tingze Feng, Meijun Chen, Shanshan Liu, Xiaocong Fu, Tianzhi Wu, Erqiang Hu and Guangchuang Yu*

*correspondence: Guangchuang Yu <gcyu1@smu.edu.cn>

1 Analysis of 16s rDNA dataset about 43 pediatric CD stool samples from iHMP

Here, we use the 43 pediatric IBD stool samples as example, which were obtained from the Integrative Human Microbiome Project Consortium (iHMP) (Research Network Consortium 2014).

1.1 Importing the output of dada2

The datasets were downloaded from web[1]. It contains `ibd_asv_table.txt`, which is feature table (*row features* X *column samples*), `ibd_meta.csv` (metadata file of samples), and `ibd_taxa.txt` (the taxonomic annotation of features). In the session, we use *mp_import_dada2* of *MicrobiotaProcess* to import the dataset, and return a *MPSE* object.

```
library(MicrobiotaProcess)
otuda <- read.table("./data/IBD_data/ibd_asv_table.txt", header=T,
                    check.names=F, comment.char="", row.names=1, sep="\t")
# building the output format of removeBimeraDenovo of dada2
otuda <- data.frame(t(otuda), check.names=F)
sampleda <- read.csv("./data/IBD_data/ibd_meta.csv", row.names=1, comment.char="")
taxda <- read.table("./data/IBD_data/ibd_taxa.txt", header=T,
                    row.names=1, check.names=F, comment.char="")
# the feature names should be the same with rownames of taxda.
taxda <- taxda[match(colnames(otuda), rownames(taxda)),]
mpse <- mp_import_dada2(seqtab = otuda, taxatab = taxda, sampleda = sampleda)
# view the reads depth of samples and the prevalence of the OTUs. In this example,
# mpse %>% mp_extract_assay(.abundant=Abundance) %>% rowSums() %>% sort %>% head(100)
# mpse %>% mp_extract_assay(.abundant=Abundance) %>% colSums() %>% sort %>% head()
# Or
# head(sort(rowSums(assay(mpse, "Abundance"))), 100)
# head(sort(colSums(assay(mpse, "Abundance"))))
# In this example, we can find some OTUs have very low frequency in the samples.
# and some taxonomy are unreasonable, for example, the probability of chloroplasts
# in the intestine should be low. We can also remove the features.
mpse2 <- mpse %>%
         dplyr::filter(!Phylum %in% c("p__un_k__Bacteria", "p__Chloroflexi") &
                       !Class %in% "c__Chloroplast" &
                       !Family %in% "f__mitochondria"
         ) %>%
         mp_filter_taxa(.abundance = Abundance, min.abun = 1, min.prop = 0.1)
mpse2
```

```
## # A MPSE-tibble (MPSE object) abstraction: 9,890 x 11
## # OTU=230 | Samples=43 | Assays=Abundance | Taxanomy=Kingdom, Phylum, Class, Order, Family, Genus, Species
##    OTU    Sample Abundance Group Kingdom Phylum Class Order Family Genus Species
##    <chr>  <chr>      <int> <chr> <chr>   <chr>  <chr> <chr> <chr>  <chr> <chr>
##  1 OTU_2~ S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__A~ f__Ac~ g__A~ s__un_~
##  2 OTU_5~ S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__A~ f__Ac~ g__A~ s__un_~
##  3 OTU_7~ S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__A~ f__Mi~ g__R~ s__muc~
##  4 OTU_42 S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__B~ f__Bi~ g__B~ s__ado~
```

[1] https://www.microbiomeanalyst.ca/MicrobiomeAnalyst/resources/data/ibd_data.zip

```
##  5 OTU_1~ S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__B~ f__Bi~ g__B~ s__un_~
##  6 OTU_1~ S2067~         0 CD    k__Bac~ p__Ac~ c__A~ o__B~ f__Bi~ g__B~ s__un_~
##  7 OTU_3~ S2067~         0 CD    k__Bac~ p__Ac~ c__C~ o__C~ f__Co~ g__A~ s__un_~
##  8 OTU_1~ S2067~         0 CD    k__Bac~ p__Ac~ c__C~ o__C~ f__Co~ g__C~ s__aer~
##  9 OTU_3~ S2067~         0 CD    k__Bac~ p__Ac~ c__C~ o__C~ f__Co~ g__E~ s__len~
## 10 OTU_1~ S2067~         0 CD    k__Bac~ p__Ba~ c__B~ o__B~ f__[O~ g__O~ s__un_~
## # ... with 9,880 more rows
```

1.2 Other import functions

MicrobiotaProcess also presents some other import functions S1 to parse the output of the upstream pipelines. In addition, some common object of R can also be converted to *MPSE* object, such as *phyloseq* (McMurdie 2013), *SummarizedExperiment* (Morgan et al. 2021), *TreeSummarizedExperiment* (Huang et al. 2021), *biom* (McMurdie and Paulson 2021) (output of *biomformat* by *read_biom*) 2.1.

Table S1: List of import functions provided by MicrobiotaProcess

Package	Import Function	Description
MicrobiotaProcess	mp_import_qiime2	Import function to load the output of qiime2
	mp_import_qiime	Import function to read the now legacy-format QIIME OTU table (tsv format)
	mp_import_metaphlan	Import function to read the output of MetaPhlAn

1.3 alpha diversity analysis

1.3.1 rarefaction visualization

Rarefaction, based on sampling technique, was used to compensate for the effect of sample size on the number of units observed in a sample. *MicrobiotaProcess* provided *mp_cal_rarecurve* and *mp_plot_rarecurve* to calculate and plot the curves.

```r
library(MicrobiotaProcess)
library(patchwork)
cols <- c("orange", "deepskyblue")
mpse2 %<>%
    mp_rrarefy(.abundance=Abundance) %>%
    mp_cal_rarecurve(.abundance=RareAbundance, chunks=500)

p_rare <- mpse2 %>%
          mp_plot_rarecurve(
            .rare = RareAbundanceRarecurve,
            .alpha = c(Observe, Chao1, ACE),
          ) +
          theme(
            legend.key.width = unit(0.3, "cm"),
            legend.key.height = unit(0.3, "cm"),
            legend.spacing.y = unit(0.01,"cm"),
            legend.text = element_text(size=4)
          )

prare1 <- mpse2 %>%
          mp_plot_rarecurve(
            .rare = RareAbundanceRarecurve,
            .alpha = c(Observe, Chao1, ACE),
            .group = Group
          ) +
          scale_fill_manual(values = cols)+
          scale_color_manual(values = cols)+
          theme_bw()+
          theme(
            axis.text=element_text(size=8), panel.grid=element_blank(),
            strip.background = element_rect(colour=NA,fill="grey"),
            strip.text.x = element_text(face="bold")
          )

prare2 <- mpse2 %>%
          mp_plot_rarecurve(
            .rare = RareAbundanceRarecurve,
            .alpha = c(Observe, Chao1, ACE),
            .group = Group,
            plot.group = TRUE
          ) +
          scale_color_manual(values = cols)+
          scale_fill_manual(values = cols) +
          theme_bw()+
          theme(
            axis.text=element_text(size=8), panel.grid=element_blank(),
            strip.background = element_rect(colour=NA,fill="grey"),
            strip.text.x = element_text(face="bold")
          )

(p_rare / prare1 / prare2) + patchwork::plot_annotation(tag_levels="A")
```

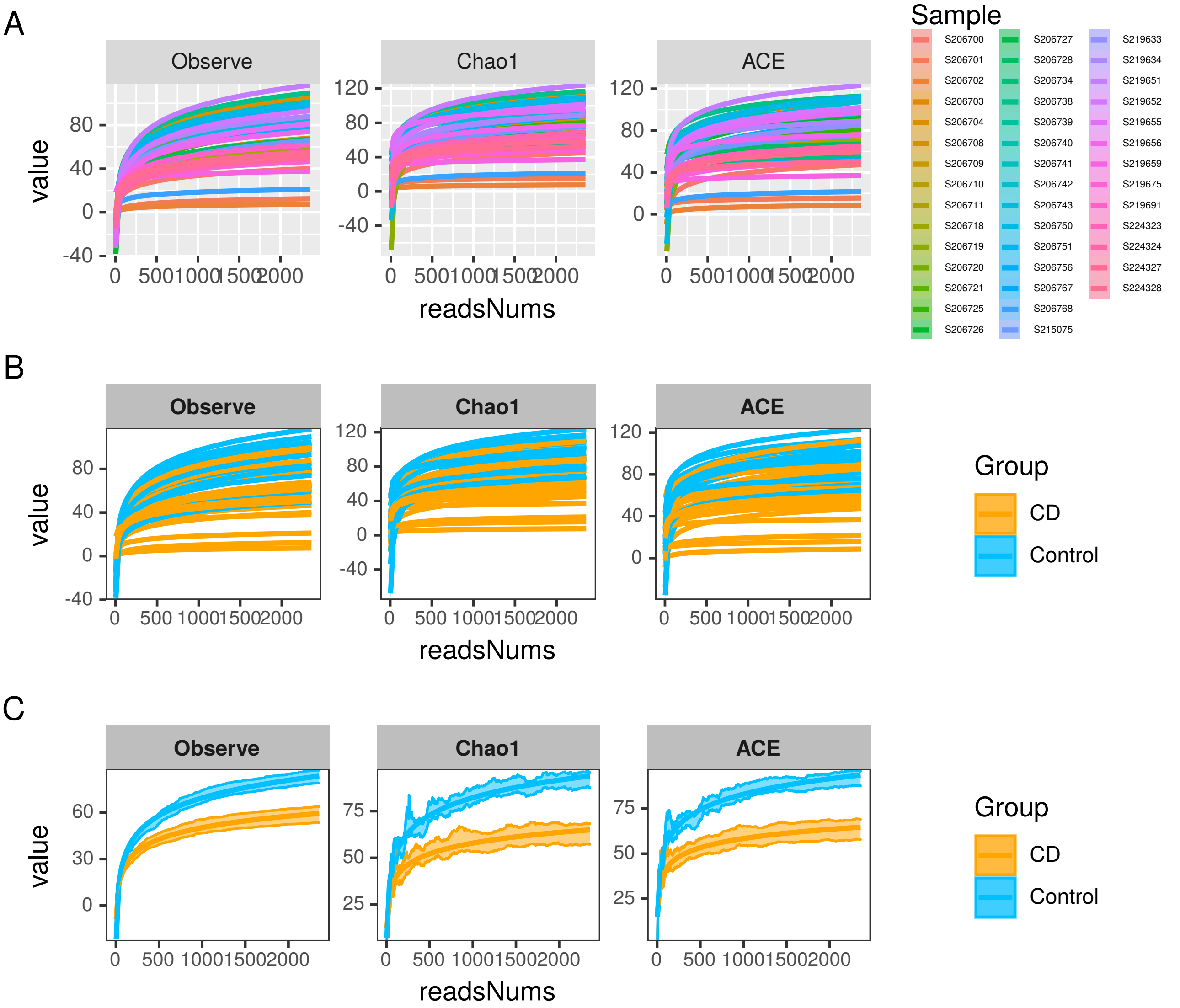


Fig. S1: **This exampls show *MicrobiotaProcess* provided *mp_cal_rarecurve* and *mp_plot_rarecurve* to calculate and visualize the rarefaction curve.** The horizontal coordinate represents the sequencing depth of samples, the vertical coordinate shows the Alpha diversity index (such as Observe OTU, Chao1 and ACE). The *mp_plot_rarecurve* provides three types of visualization. (A) the rarefaction curve for each sample. (B) the rarefaction curve for each sample with colored group (specified *.group* argument in *mp_plot_rarecurve*). (C) the rarefaction curve for each group with standard error of the mean (specified *.group* argument and *plot.group=TRUE* in *mp_plot_rarecurve*)

Since the curves in each sample were near saturation, the sequencing data were great enough with very few new species undetected

1.3.2 Calculation and different analysis of alpha index

Alpha index can evaluate the richness and abundance of microbial communities. *MicrobiotaProcess* provides *mp_cal_alpha* to calculate alpha index. Six common diversity measures (*Observe*, *Chao1*, *ACE*, *Shannon*, *Simpson*, *Pielou*) are supported. And the different groups of samples can be tested and visualize by *mp_plot_alpha*. This following example shows how to use *mp_cal_alpha* and *mp_plot_alpha* of *MicrobiotaProcess* to analysis the alpha diversity of the community. The *RareAbundance* is rarefied (default), which will be used to calculate the alpha diversity index, users can specified the *force=TRUE* of *mp_cal_alpha* to calculated the index if the abundance is not be rarefied (2.3.1).

```
library(MicrobiotaProcess)
mpse2 %<>% mp_cal_alpha(.abundance = RareAbundance)
p_alpha <- mpse2 %>%
           mp_plot_alpha(
              .alpha = c(Observe, Chao1, ACE, Shannon, Simpson, Pielou),
              .group = Group,
           ) +
           scale_fill_manual(values=cols)+
           scale_color_manual(values=cols) +
           theme(legend.position="none",
                 strip.background = element_rect(colour=NA, fill="grey"))
p_alpha
```

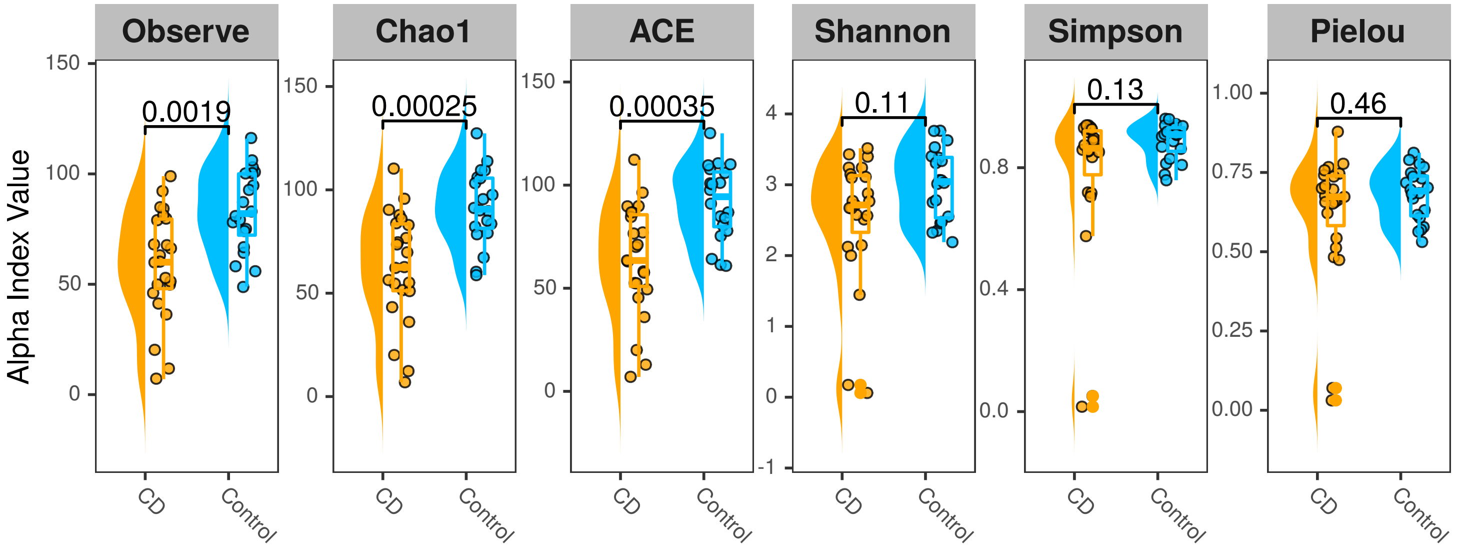


Fig. S2: **The raincloud plot of alpha diversity index** The horizontal coordinate represents each group (by *.group* argument of *mp_plot_alpha*), the vertical coordinate represents the alpha diversity index.

1.4 Taxonomy composition analysis

1.4.1 Statistics and visualization of specific levels

MicrobiotaProcess presents the *mp_cal_abundance* and *mp_plot_abundance* for the calculation and visualization of composition of microbial communities. After the *mp_cal_abundance* done, you can get the abundance of specific levels of class by *mp_extract_abundance* 1.5.4.

```
library(ggplot2)
library(MicrobiotaProcess)
# The relative abundance of all taxonomy for samples will be calculated
mpse2 %<>% mp_cal_abundance(.abundance = RareAbundance)
# The relative abundance of all taxonomy for group will be calculated
mpse2 %<>% mp_cal_abundance(.abundance = RareAbundance, .group = Group)
# The 30 most abundant taxonomy will be visualized.
pclass <- mpse2 %>%
          mp_plot_abundance(
             .abundance = RareAbundance,
             .group = Group,
             taxa.class = Class,
             topn = 30
          ) +
          xlab(NULL) +
          ylab("relative abundance (%)") +
          theme(
             legend.key.width = unit(0.3, "cm"),
             legend.key.height = unit(0.3, "cm")
          ) +
          xlab(NULL) +
          ylab("relative abundance (%)") +
          theme(
             legend.key.width = unit(0.3, "cm"),
             legend.key.height = unit(0.3, "cm"),
             legend.text = element_text(size=6)
          )
pclass
```

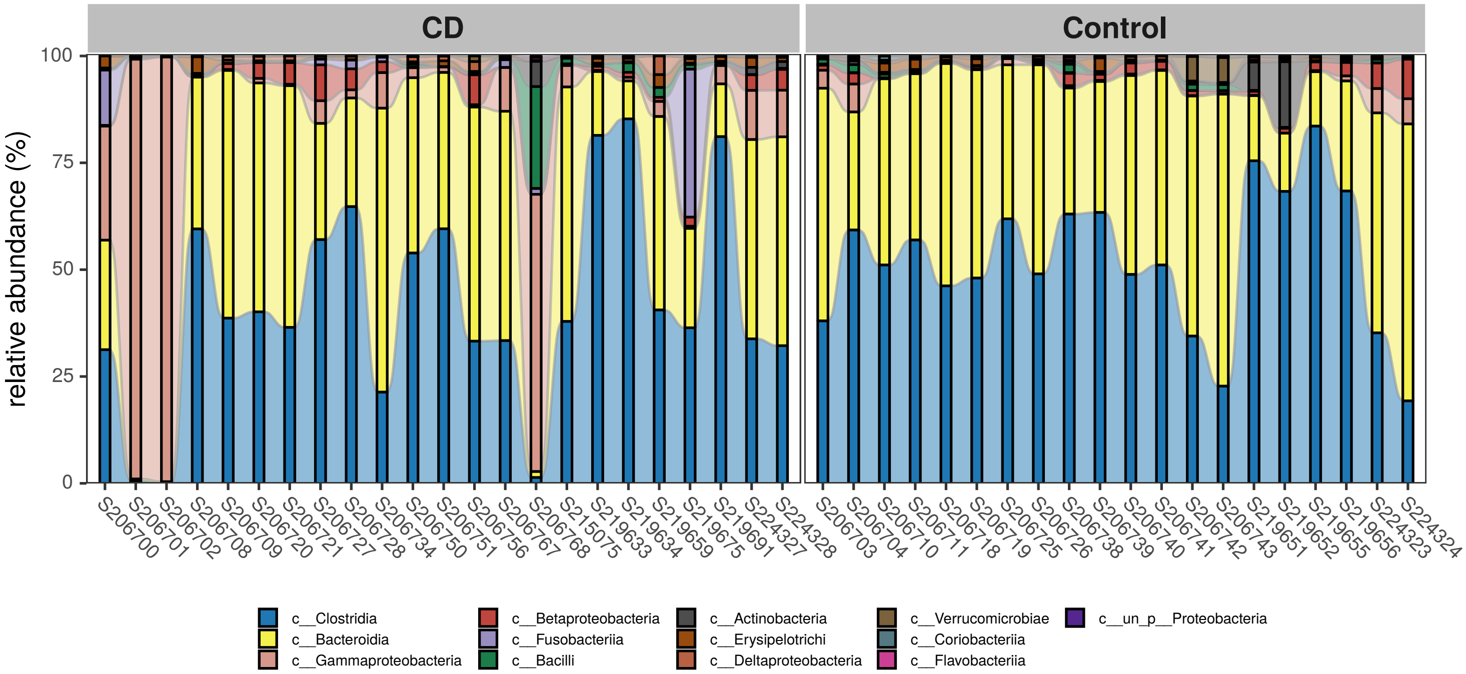


Fig. S3: **The relative abundance of each sample in *class* level**

The relative abundance of groups also can be visualized by providing *.group* argument and setting *plot.group=TRUE* in the *mp_plot_abundance*. If you want to view the raw abundance (count or others) of taxa, you can set the *relative* parameter of *mp_plot_abundance* to *FALSE*.

```
# Show the abundance in different groups.
fclass <- mpse2 %>%
          mp_plot_abundance(
             .abundance = RareAbundance,
             .group = Group,
             taxa.class = Class,
             topn = 30,
             plot.group = TRUE
          ) +
          xlab(NULL) +
          ylab("relative abundance (%)") +
          theme(legend.position = "none")

pclass2 <- mpse2 %>%
          mp_plot_abundance(
             .abundance = RareAbundance,
             .group = Group,
             relative = FALSE,
             taxa.class = Class,
             topn = 30
          ) +
          xlab(NULL) +
          ylab("count reads") +
          theme(legend.key.width = unit(0.3, "cm"),
                legend.key.height = unit(0.3, "cm"),
                 legend.text = element_text(size=6)
          )

aplot::plot_list(pclass2, fclass, widths=c(10, 1), tag_levels = "A")
```

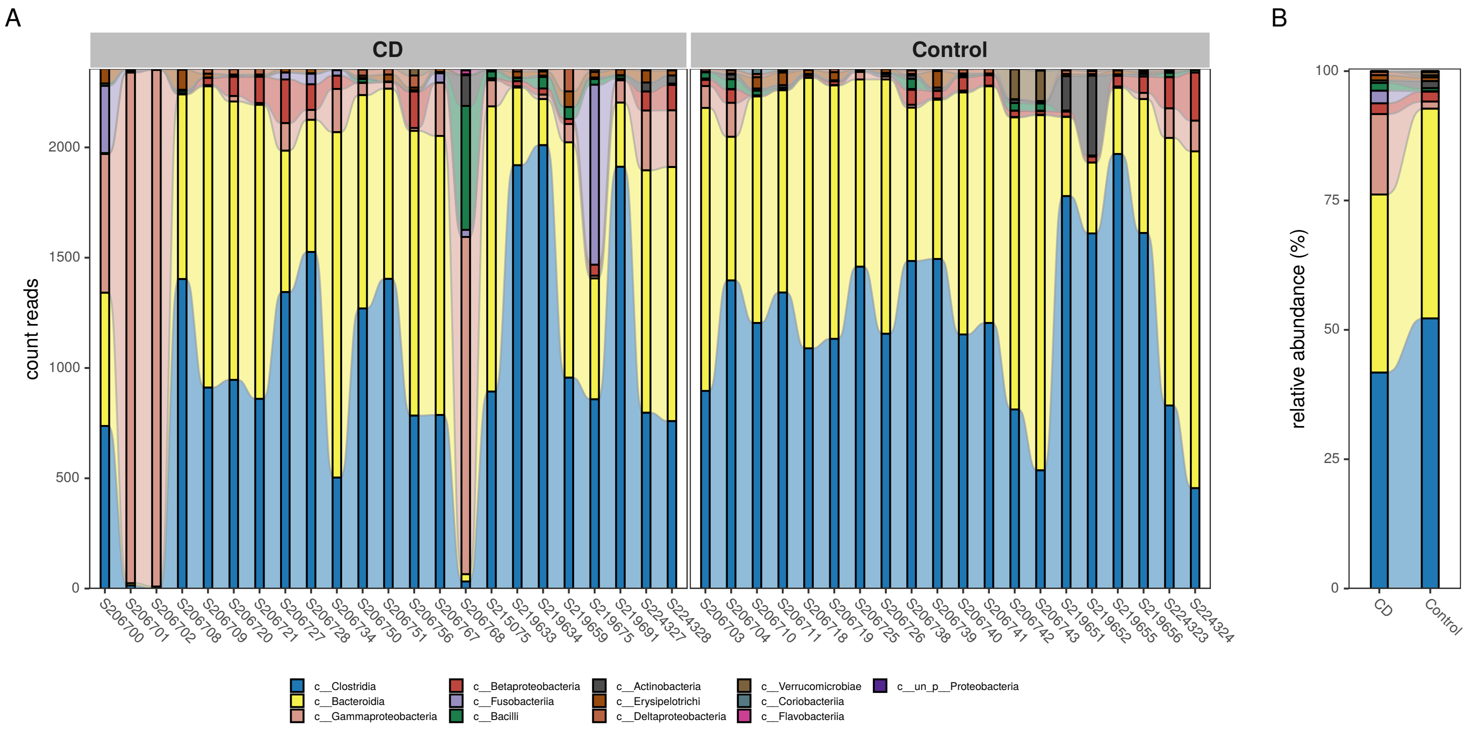


Fig. S4: **This example show how to displayed the abundance (count or other) of sample and the relative abundance of groups**. The Abundance (count by rarefied) of each sample (A) and the relative abundance of group (B), these results show the *Gammaproteobacteria* of *CD* group might be more abundant than the *control* group.

The abundance of features also can be visualized by `mp_plot_abundance` with heatmap plot by setting `geom="heatmap"`.

```
hclass1 <- mpse2 %>%
          mp_plot_abundance(
             .abundance = RareAbundance,
             .group = Group,
             taxa.class = Class,
             topn = 30,
             geom = "heatmap"
          ) %>%
          set_scale_theme(
            x = list(scale_fill_viridis_c(option = "H"),
                     theme(
                       axis.text.x = element_text(size = 6),
                       axis.text.y = element_text(size = 7),
                       legend.title = element_text(size = 7),
                       legend.text = element_text(size = 5),
                       legend.key.width = unit(0.3, "cm"),
                       legend.key.height = unit(0.3, "cm")
                     )
                ),
            aes_var = RelRareAbundance
          ) %>%
          set_scale_theme(
            x = list(scale_fill_manual(values = cols),
                     theme(
                       legend.key.height = unit(0.3, "cm"),
                       legend.key.width = unit(0.3, "cm"),
                       legend.spacing.y = unit(0.02, "cm"),
                       legend.text = element_text(size = 7),
                       legend.title = element_text(size = 9)
                     )
                ),
            aes_var = Group
          )

hclass2 <- mpse2 %>%
          mp_plot_abundance(
             .abundance = RareAbundance,
             .group = Group,
             taxa.class = Class,
             topn = 30,
             geom = 'heatmap',
             relative = FALSE
          ) %>%
          set_scale_theme(
            x = list(scale_fill_viridis_c(option = "H"),
                     theme(
                       axis.text.x = element_text(size = 6),
                       axis.text.y = element_text(size = 7),
                       legend.title = element_text(size = 7),
                       legend.text = element_text(size = 5),
                       legend.key.width = unit(0.3, "cm"),
                       legend.key.height = unit(0.3, "cm")
                     )
                ),
            aes_var = RareAbundance
          ) %>%
          set_scale_theme(
            x = list(scale_fill_manual(values = cols),
                     theme(
                       legend.key.height = unit(0.3, "cm"),
                       legend.key.width = unit(0.3, "cm"),
                       legend.spacing.y = unit(0.02, "cm"),
                       legend.text = element_text(size = 7),
                       legend.title = element_text(size = 9)
                     )
                ),
            aes_var = Group
          )

p <- aplot::plot_list(hclass1, hclass2, nrow = 1, tag_levels = "A")
p
```

1.4.2 Venn or Upset plot

The Venn or UpSet plot can help us to obtain the difference between groups in overview. MicrobiotaProcess provides `mp_cal_venn` (`mp_plot_venn`) and `mp_cal_upset` (`mp_plot_upset`) to perform the Venn and Upset analysis.

```
mpse2 %<>%
    mp_cal_venn(
      .abundance = RareAbundance,
      .group = Group
    )

venn_p <- mpse2 %>%
    mp_plot_venn(
      .group = Group,
```

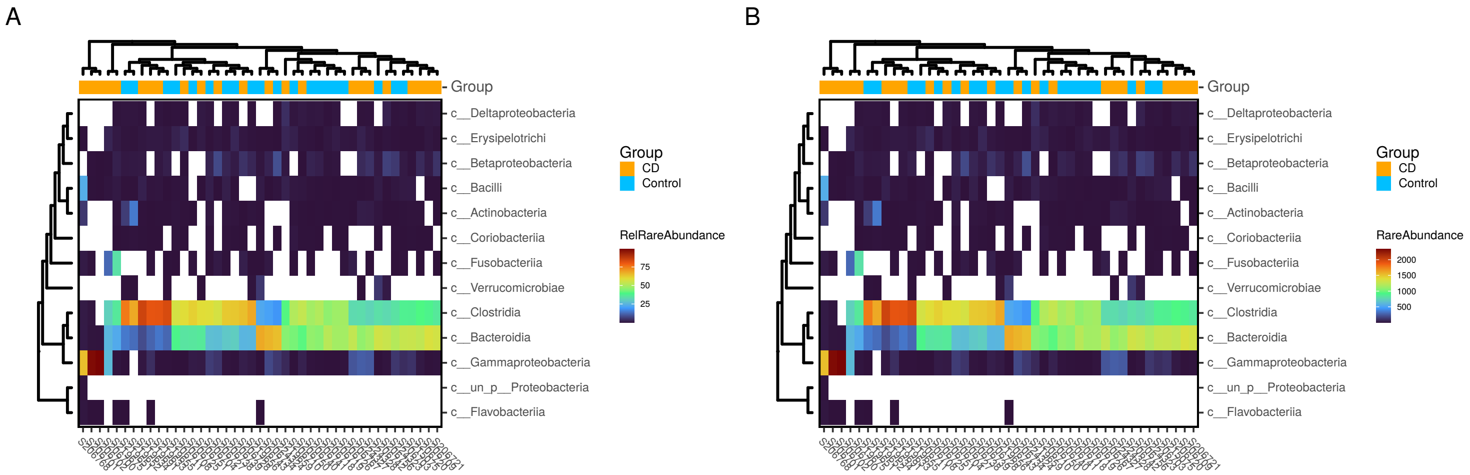


Fig. S5: **The heatmap of abundance for each sample in *class* level.** The color (continuous) of heatmap represents the abundance of taxon, the color of bar represents the group name of sample, the horizontal coordinate represents the sample, and the vertical coordinate represents the taxon.

```
      set_size = 2.5,
      label_size = 2,
      edge_size = 2.5
    ) +
    scale_colour_manual(values = cols) +
    scale_fill_viridis_c(guide = guide_colorbar(barwidth=.3, barheight=2)) +
    theme(
      legend.title = element_text(size = 8),
      legend.text = element_text(size = 6)
    )

mpse2 %<>%
    mp_cal_upset(
      .abundance = RareAbundance,
      .group = Group
    )

upset_p <- mpse2 %>%
    mp_plot_upset(
      .group = Group
    ) +
    theme_bw() +
    theme(
      plot.background = element_blank(),
      panel.border = element_blank(),
      panel.grid = element_blank(),
      axis.line.x.bottom = element_line(size = .5),
      axis.line.y.left = element_line(size = .5)
    ) +
    ggupset::theme_combmatrix(
      combmatrix.label.extra_spacing = 40
    )

library(ggpp)
p.up.venn <- upset_p +
             ggpp::annotate(
               "plot_npc",
               npcx = "right",
               npcy = "top",
               label = venn_p,
               vp.width = 0.6,
               vp.height = 0.4
             )
p.up.venn
```

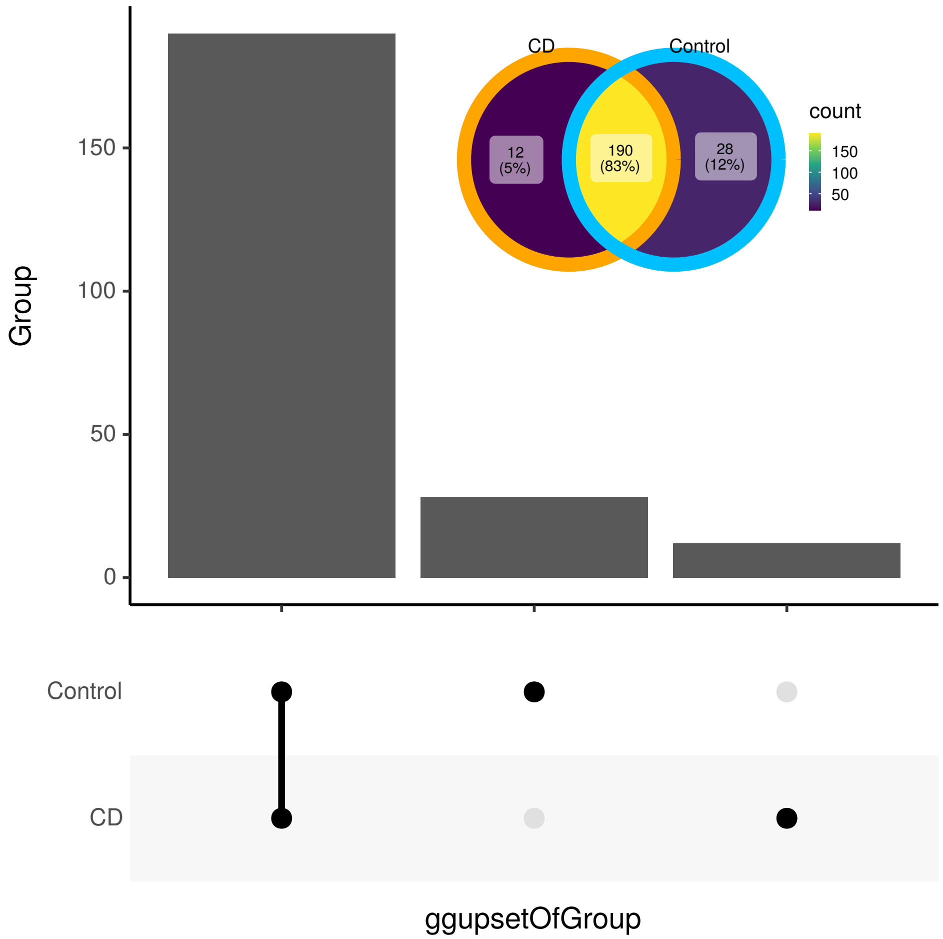


Fig. S6: **The venn diagram and upset plot for groups in OTU/ASV level**

1.5 beta analysis

1.5.1 PCA analysis

PCA (Principal component analysis) and PCoA (Principal Coordinate Analysis) are general statistical procedures to compare dissimilarity of samples. And PCoA can based on the phylogenetic or count-based distance metrics, such as Bray-Curtis, Jaccard, Unweighted-UniFrac and weighted-UniFrac. MicrobiotaProcess presents the mp_cal_dist, mp_cal_pca, mp_cal_pcoa, mp_cal_dca, mp_cal_nmds, mp_cal_cca, mp_cal_rda, mp_adonis, mp_anosim, mp_mrpp, mp_envfit and mp_mantel for the analysis.

```
library(MicrobiotaProcess)
library(patchwork)
# hellinger transform
mpse2 %<>%
    mp_decostand(
        .abundance = Abundance,
        method = "hellinger"
    )

mpse2 %<>% mp_cal_pca(.abundance = hellinger)
# Visulizing the result
pcaplot1 <- mpse2 %>%
            mp_plot_ord(
              .ord = pca,
              .group = Group,
              .starshape = Group,
              .size = Observe
            ) +
            scale_fill_manual(values = cols) +
            scale_size_continuous(
              range = c(1, 3),
              guide = guide_legend(override.aes = list(starshape = 15))
            ) +
            theme(
              legend.key.width = unit(0.3, "cm"),
              legend.key.height = unit(0.3, "cm"),
              legend.text = element_text(size = 6),
              legend.title = element_text(size = 7)
            )
# .dim = c(1, 3) to show the first and third principal components.
pcaplot2 <- mpse2 %>%
            mp_plot_ord(
              .ord = pca,
              .dim = c(1, 3),
              .group = Group,
              .starshape = Group,
              .size = Observe
            ) +
            scale_fill_manual(values = cols) +
            scale_size_continuous(
              range = c(1, 3),
              guide = guide_legend(override.aes = list(starshape = 15))
            ) +
            theme(
              legend.key.width = unit(0.3, "cm"),
              legend.key.height = unit(0.3, "cm"),
              legend.text = element_text(size = 6),
              legend.title = element_text(size = 7)
            )

(pcaplot1 | pcaplot2) + plot_annotation(tag_levels = "A")
```

1.5.2 PCoA analysis

```
# distmethod
# "unifrac",  "wunifrac", "manhattan", "euclidean", "canberra", "bray", "kulczynski" ...(vegdist, dist)
mpse2 %<>%
    mp_cal_dist(
      .abundance = hellinger,
      distmethod = "bray"
    )

# PCoA analysis
mpse2 %<>%
    mp_cal_pcoa(
      .abundance = hellinger,
      distmethod = "bray"
    )
pcoaplot1 <- mpse2 %>%
             mp_plot_ord(
               .ord = pcoa,
```

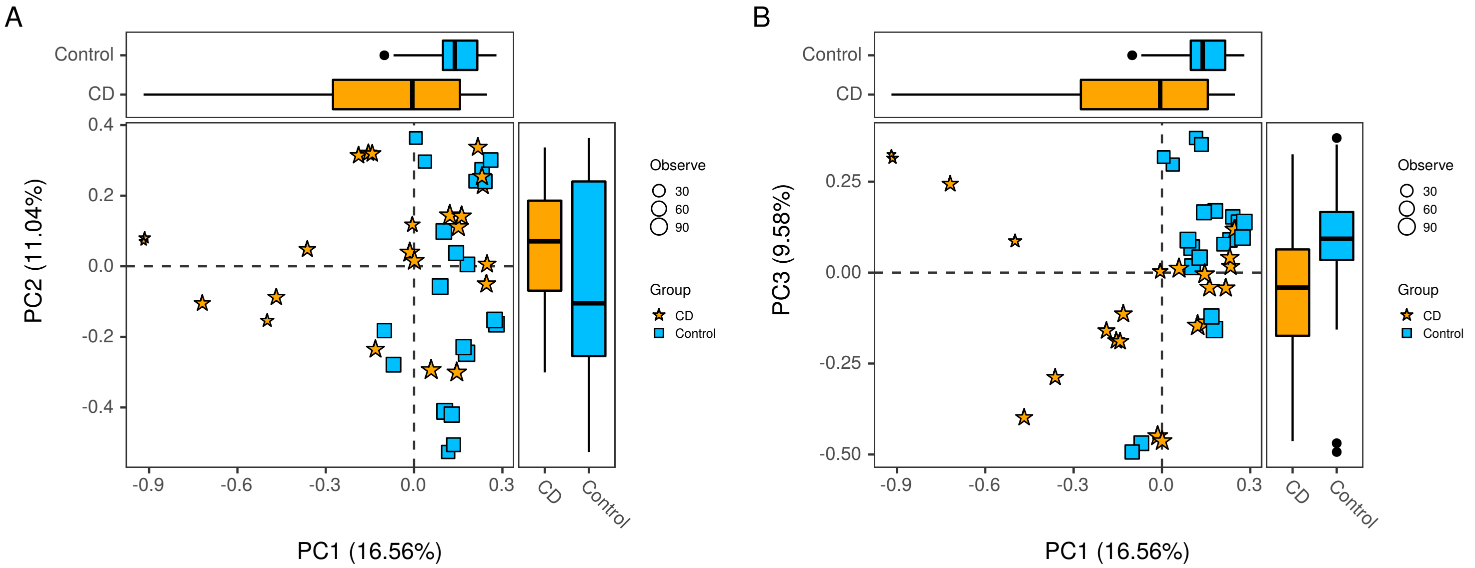


Fig. S7: **The PCA plot of the community**. Each point represents one sample, the size of point represents the observe OTU of the sample. The color of point represents the group name of the sample, based on the first and second component (A), based on the first and third component (B).

```
               .group = Group,
               .starshape = Group,
               .color = Group,
               .size = Observe,
               ellipse = TRUE
            ) +
            scale_color_manual(
               values = cols,
               guide = "none"
            ) +
            scale_fill_manual(values = cols) +
            scale_size_continuous(
               range = c(1, 3),
               guide = guide_legend(override.aes = list(starshape = 15))
            ) +
            theme(
               legend.key.width = unit(0.3, "cm"),
               legend.key.height = unit(0.3, "cm"),
               legend.text = element_text(size=6),
               legend.title = element_text(size=7)
            )
# first and third principal co-ordinates
pcoaplot2 <- mpse2 %>%
             mp_plot_ord(
               .ord = pcoa,
               .group = Group,
               .starshape = Group,
               .color = Group,
               .size = Observe,
               ellipse = TRUE,
               .dim = c(1, 3)
             ) +
             scale_color_manual(
               values = cols,
               guide = "none"
             ) +
             scale_fill_manual(
               values = cols
             ) +
             scale_size_continuous(
               range = c(1, 3),
               guide = guide_legend(override.aes = list(starshape = 15))
             ) +
             theme(
               legend.key.width = unit(0.3, "cm"),
               legend.key.height = unit(0.3, "cm"),
               legend.text = element_text(size = 6),
               legend.title = element_text(size = 7)
             )
(pcoaplot1 | pcoaplot2) + plot_annotation(tag_levels = "A")
```

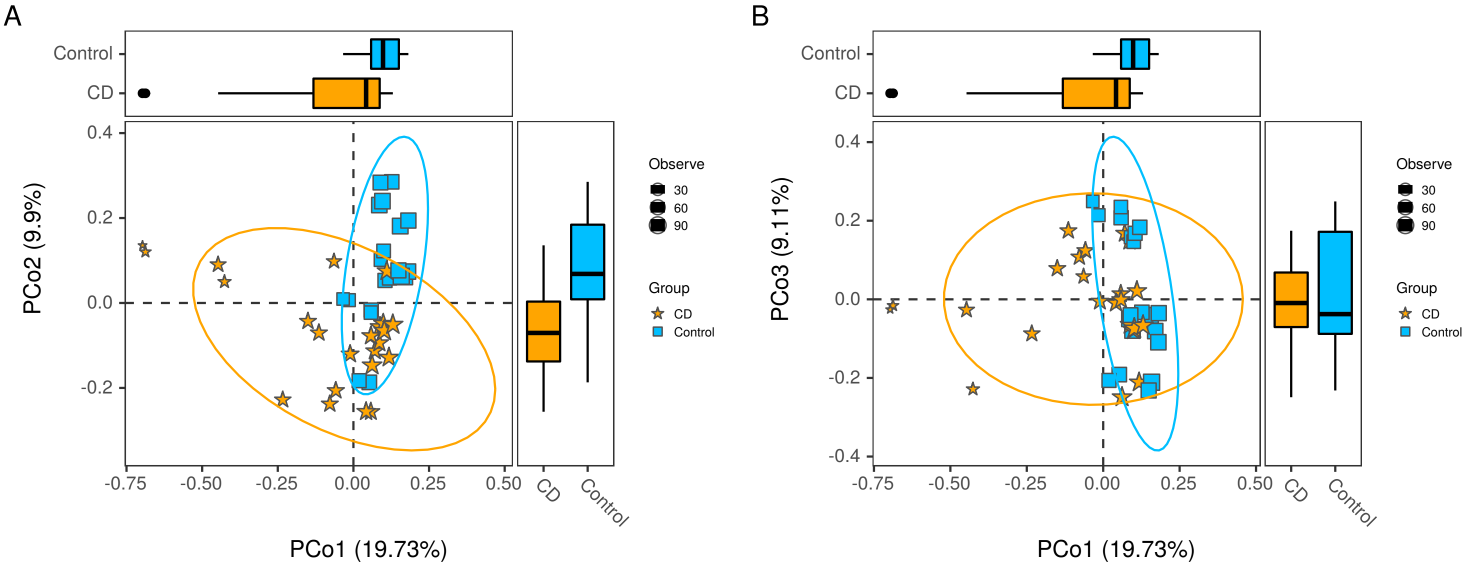


Fig. S8: **The PCoA plot based on Bray-Curtis distance.**

The result of distance between the samples also can be visualized by `mp_plot_dist` with heatmap or boxplot.

```
pdist1 <- mpse2 %>%
          mp_plot_dist(
            .distmethod = bray,
            .group = Group
          ) %>%
          set_scale_theme(
            x = scale_fill_manual(
                  values=cols,
                  guide = guide_legend(
                            keywidth = 0.5,
                            keyheight = 0.5,
                            label.theme=element_text(size=6)
                  )
                ),
            aes_var = Group
          ) %>%
          set_scale_theme(
            x = list(scale_size_continuous(range = c(1, 3)),
                  scale_color_viridis_c(option = "H"),
                  theme(
                    legend.key.width = unit(0.3, "cm"),
                    legend.text = element_text(size = 6),
                    legend.title = element_text(size = 7)
                  )
                ),
              aes_var = bray
            )

pdist2 <- mpse2 %>%
          mp_plot_dist(
            .distmethod = bray,
            .group = Group,
            group.test = TRUE
          ) +
          scale_color_manual(
            values = c("orange", "#00A08A", "deepskyblue")
          ) +
          scale_fill_manual(
            values = c("orange", "#00A08A", "deepskyblue")
          )
aplot::plot_list(pdist1, pdist2, widths = c(3, 1), nrow=1, tag_levels = "A")
```

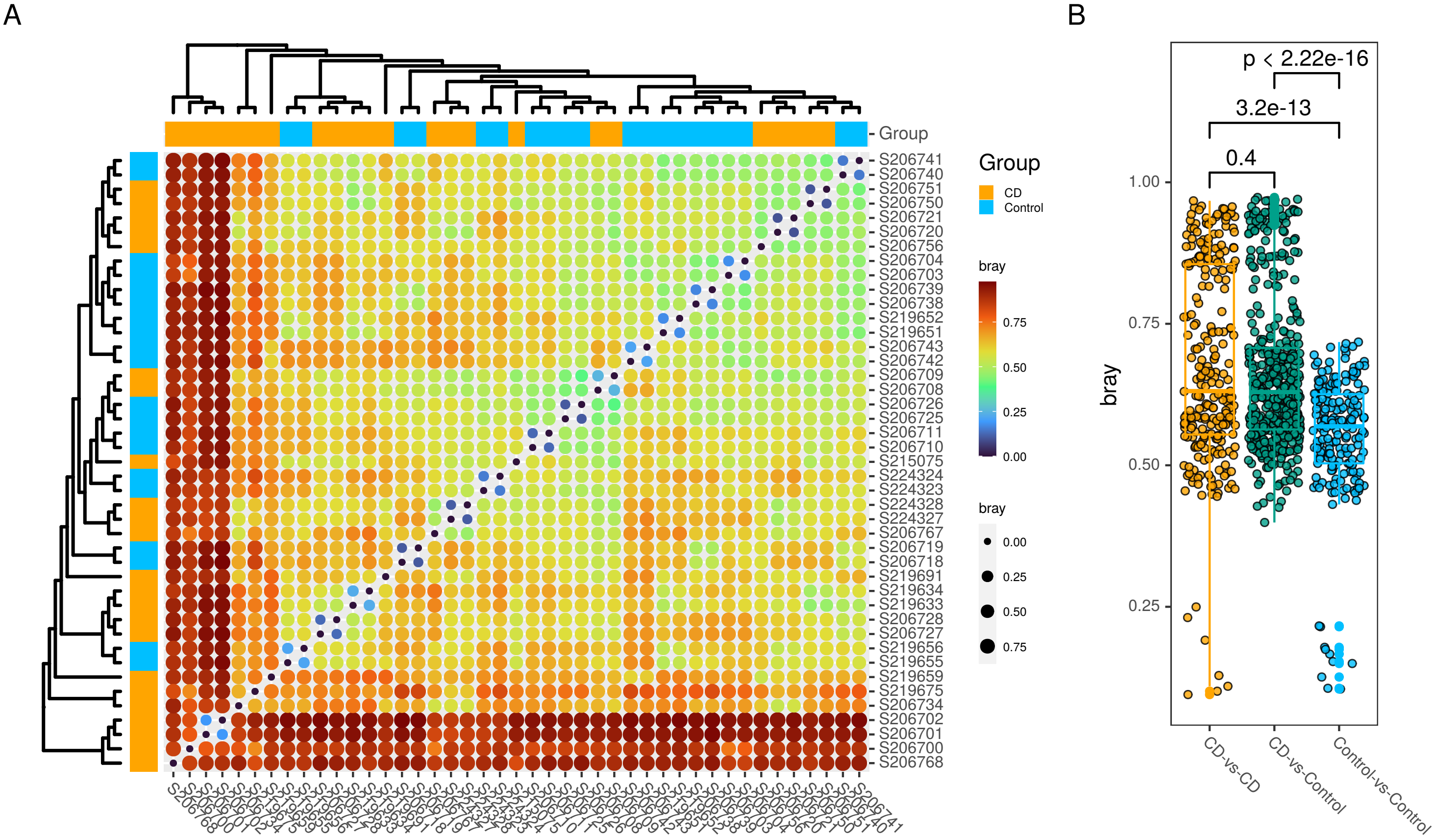


Fig. S9: **The distance heatmap and the boxplot for each sample**. The size and color of the heatmap represent the distance of each sample, the color of bar represent the group of sample (A). The boxplot represent the distance pairs of sample among the group, it show the dissimilarity of sample between the *control* and *CD* is significant, which is consistent with the result of Permutational Multivariate Analysis of Variance 1.5.3.

1.5.3 Permutational Multivariate Analysis of Variance

We also can perform the Permutational Multivariate Analysis of Variance using `mp_adonis` wrapping the `adonis` of `vegan` (Oksanen et al. 2020).

```
mpse2 %<>% mp_adonis(
            .abundance = hellinger,
            distmethod = "bray",
            .formula = ~Group,
            permutation = 9999,
            action = "add")
```

```
## The result of adonis has been saved to the internal attribute !

## It can be extracted using this-object %>% mp_extract_internal_attr(name='adonis')
```

```
mpse2 %>%
    mp_extract_internal_attr(name=adonis) %>%
    mp_fortify()
```

```
## The object contained internal attribute: PCA PCoA ADONIS

## # A tibble: 3 x 7
##   factors      Df SumsOfSqs MeanSqs F.Model     R2 `Pr(>F)`
##   <chr>     <dbl>     <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
## 1 Group         1     0.789   0.789    3.88 0.0864   0.0001
## 2 Residuals    41     8.34    0.203   NA    0.914   NA
## 3 Total        42     9.12   NA       NA    1       NA
```

From the result, we found the `pvalue` of the analysis of `adonis` is smaller than 0.05 for the `Group`, meaning the dissimilarity of samples between the `Group` is significant, which is consistent with the 1.5.2.

1.5.4 hierarchical cluster analysis of samples

`beta diversity` metrics can assess the differences between microbial communities. It can be visualized with `PCA` or `PCoA`, this can also be visualized with hierarchical clustering based on ggplot2 (Wickham 2011), ggtree (Yu et al. 2017) and ggtreeExtra (Xu et al. 2021)

```
library(ggplot2)
library(MicrobiotaProcess)
library(ggtree)
library(ggtreeExtra)
mpse2 %<>%
    mp_cal_clust(
      .abundance = hellinger,
      distmethod = "bray",
      action = "add"
    )
hcsample <- mpse2 %>%
            mp_extract_internal_attr(name=SampleClust)
# rectangular layout + relative abundance of phyla
phy.tb <- mpse2 %>%
          mp_extract_abundance(
            taxa.class = Phylum,
            topn = 30
          ) %>%
          tidyr::unnest(cols=RareAbundanceBySample) %>%
          dplyr::rename(Phyla="label")

cplot1 <- ggtree(
            hcsample,
            layout = "rectangular"
          ) +
          geom_treescale(fontsize = 2) +
          geom_tippoint(mapping=aes(color=Group)) +
          geom_fruit(
            data = phy.tb,
            geom = geom_col,
            mapping = aes(x = RelRareAbundanceBySample, y = Sample, fill = Phyla),
            orientation = "y",
            offset = 0.08,
            pwidth = 3,
            width = .6,
            axis.params = list(
              axis = "x",
              title = "The relative abundance of phyla (%)",
              title.size = 3,
              title.height = 0.04,
              text.size = 2,
              vjust = 1
            )
          ) +
          geom_tiplab(as_ylab = TRUE) +
          scale_color_manual(
            values = cols,
            guide = guide_legend(
              keywidth = .5,
              keyheight = .5,
              title.theme = element_text(size = 8),
              label.theme = element_text(size = 6)
            )
          ) +
          scale_fill_manual(
            values=c(colorRampPalette(RColorBrewer::brewer.pal(12,"Set2"))(6)),
            guide = guide_legend(
              keywidth = .5,
              keyheight = .5,
              title.theme = element_text(size = 8),
              label.theme = element_text(size = 6)
            )
          ) +
          scale_x_continuous(expand = c(0, 0.01))
cplot1
```

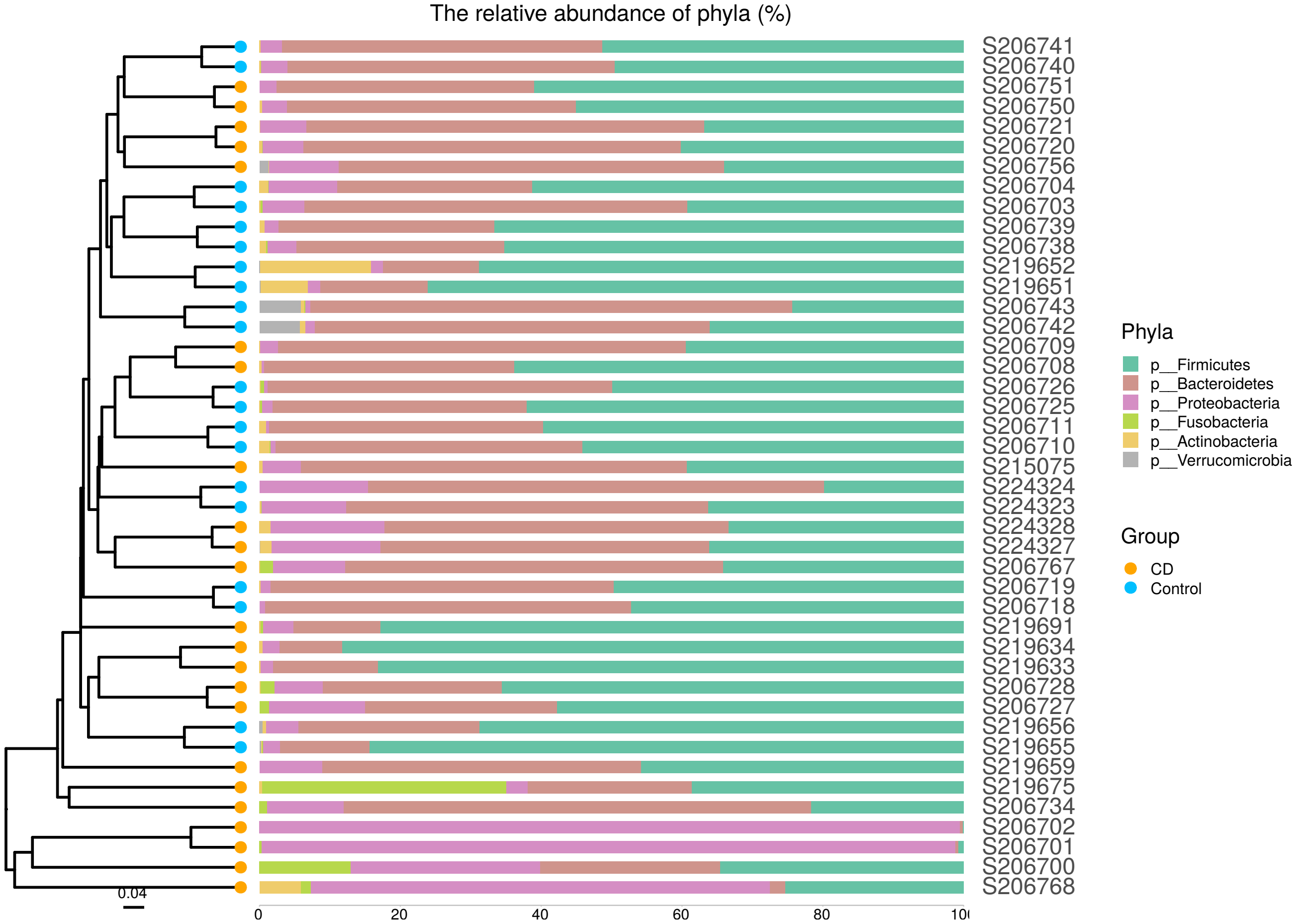


Fig. S10: **The hierarchical clustering plot of samples based on Bray-Curtis distance calculated with abundance of OTU/ASV and the relative Abundance of phyla for samples**

1.6 biomarker discovery

This package provides `mp_diff_analysis` to detect the biomarker. And the result (with `action = "get"`) can be visualized by `ggdiffbox`, `ggdiffclade`, `ggeffectsize`, `ggdifftaxbar` and `mp_plot_diff_res`, or displayed manually using `ggtree` (Yu et al. 2017) and `ggtreeExtra` (Xu et al. 2021).

```r
# for the kruskal_test and wilcox_test
library(coin)
library(MicrobiotaProcess)

# get result (diffAnalysisClass) of the different analysis with action = 'get'.
deres <- mpse2 %>%
         mp_diff_analysis(
            .abundance = RareAundance,
            .group = Group,
            first.test.method = "kruskal_test",
            filter.p = "pvalue",
            first.test.alpha = 0.05,
            strict = TRUE,
            second.test.method = "wilcox_test",
            second.test.alpha = 0.05,
            subcl.min = 3,
            subcl.test = TRUE,
            ml.method = "lda",
            ldascore = 3,
            action = "get"
         )
```

```
mpse2 <- mpse2 %>%
         mp_diff_analysis(
            .abundance = RareAundance,
            .group = Group,
            first.test.method = "kruskal_test",
            filter.p = "pvalue",
            first.test.alpha = 0.05,
            strict = TRUE,
            second.test.method = "wilcox_test",
            second.test.alpha = 0.05,
            subcl.min = 3,
            subcl.test = TRUE,
            ml.method = "lda",
            ldascore = 3,
            action = "add"
         )
```

1.6.1 visualization of different results by `ggdiffclade`

The color of discriminative taxa represent the taxa is more abundant in the corresponding group. The point size shows the negative logarithms (base 10) of pvalue. The bigger size of point shows more significant (lower pvalue), the *pvalue* was calculated in the first step test (default is *kruskal.test*).

```
diffclade_p <- ggdiffclade(
                   obj=deres,
                   alpha=0.3,
                   linewd=0.15,
                   skpointsize=0.6,
                   layout="radial",
                   taxlevel=3,
                   removeUnkown = TRUE,
                   reduce = FALSE # This argument is to remove the branch of unknown taxonomy.
               ) +
               scale_fill_manual(
                   values = cols
               ) +
               guides(color = guide_legend(
                                  keywidth = 0.1,
                                  keyheight = 0.2,
                                  order = 3,
                                  ncol=1)
               ) +
               theme(
                   panel.background = element_rect(fill=NA),
                   legend.position = "right",
                   plot.margin = margin(0,0,0,0),
                   legend.key.width = unit(0.2, "cm"),
                   legend.key.height = unit(0.2, "cm"),
                   legend.spacing.y = unit(0.02, "cm"),
                   legend.title = element_text(size=7),
                   legend.text = element_text(size=6),
                   legend.box.spacing = unit(0.02,"cm")
               )
diffclade_p
```

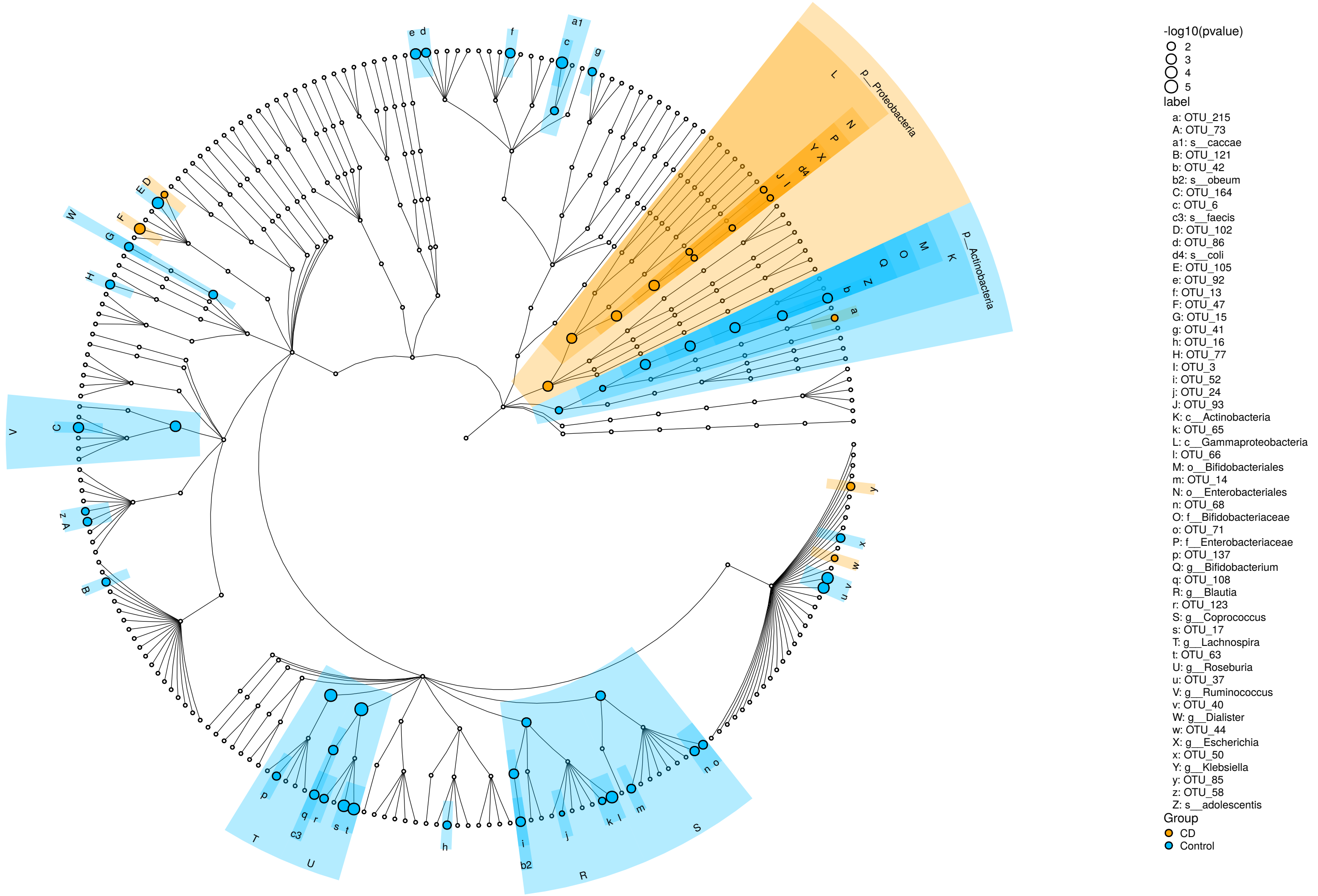


Fig. S11: **The taxa tree clade plot of different analysis result.**

We also can visualized the result with `ggtree` (Yu et al. 2017) and `ggtreeExtra` (Xu et al. 2021).

```
taxa.tree <- mpse2 %>% mp_extract_tree(type='taxatree')
p1 <- ggtree(
        taxa.tree,
          layout="radial",
          size = 0.3
      ) +
      geom_point(
        data = td_filter(!isTip),
        fill="white",
        size=1,
        shape=21
      )
# display the high light of phylum clade.
p2 <- p1 +
      geom_hilight(
        data = td_filter(nodeClass == "Phylum"),
        mapping = aes(node = node, fill = label)
      )
# display the relative abundance of features(OTU)
p3 <- p2 +
      ggnewscale::new_scale("fill") +
      geom_fruit(
         data = td_unnest(RareAbundanceBySample),
         geom = geom_star,
         mapping = aes(
                       x = fct_reorder(Sample, Group, .fun=min),
                       size = RelRareAbundanceBySample,
                       fill = Group,
                       subset = RelRareAbundanceBySample > 0
                   ),
         starshape = 13,
         starstroke = 0.25,
         offset = 0.04,
         pwidth = 1.5,
         grid.params = list(vline = TRUE, size = 0.01, linetype = 1)
      ) +
      scale_size_continuous(
         name="Relative Abundance (%)",
         range = c(1, 3)
      ) +
      scale_fill_manual(values=cols)
# display the tip labels of taxa tree
p4 <- p3 + geom_tiplab(size=2, offset=12.8)
# display the LDA of significant OTU.
p5 <- p4 +
      ggnewscale::new_scale("fill") +
      geom_fruit(
         geom = geom_col,
         mapping = aes(
                       x = LDAmean,
                       fill = Sign_Group,
                       subset = !is.na(LDAmean)
                       ),
         orientation = "y",
         offset = 0.5,
         pwidth = 1,
         axis.params = list(axis = "x",
                            title = "Log10(LDA)",
                            title.height = 0.005,
                            title.size = 2,
                            text.size = 1.8,
                            vjust = 1),
         grid.params = list(linetype = 3)
      )

# display the significant (FDR) taxonomy after kruskal.test (default)
p6 <- p5 +
      ggnewscale::new_scale("size") +
      geom_point(
         data=td_filter(!is.na(fdr)),
         mapping = aes(size = -log10(fdr),
                       fill = Sign_Group,
                       ),
         shape = 21,
      ) +
      scale_size_continuous(range=c(1, 3)) +
      scale_fill_manual(values=cols)

p6 <- p6 + theme(
           legend.key.height = unit(0.3, "cm"),
           legend.key.width = unit(0.3, "cm"),
           legend.spacing.y = unit(0.02, "cm"),
           legend.text = element_text(size = 7),
           legend.title = element_text(size = 9),
            )
p6
```

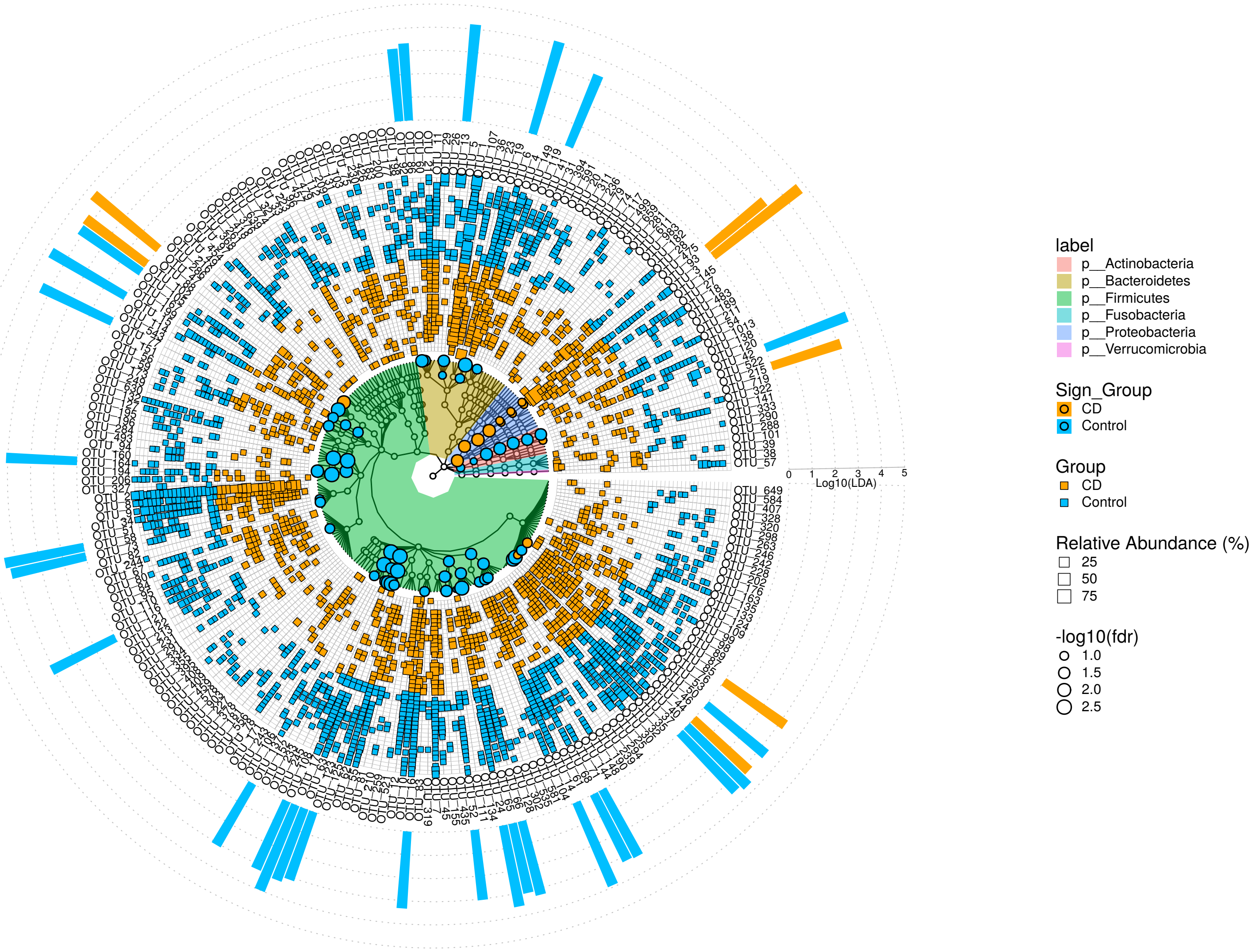


Fig. S12: **The taxa tree of the community with the relative abundance of each OTU/ASV on sample and the LDA of different OTU/ASV**. The taxa tree is built with the taxa of all samples. The high light color of taxa tree represents the phyla of the clade. The external point layer represents the relative abundance of each OTU on sample. The external bar layer represents the LDA of the different OTU. The colored points represent the different taxa, the size represents the *pvalue* or *fdr*.

1.6.2 visualization of different results by `ggdiffbox`

The left panel represents the relative abundance or abundance (according the standard_method) of biomarker, the right panel represents the confident interval of effect size (LDA or MDA) of biomarker. The bigger confident interval shows that the biomarker is more fluctuant, owing to the influence of samples number.

```r
diffbox <- ggdiffbox(obj=deres, box_notch=FALSE,
             colorlist=cols, l_xlabtext="relative abundance")
diffbox
```

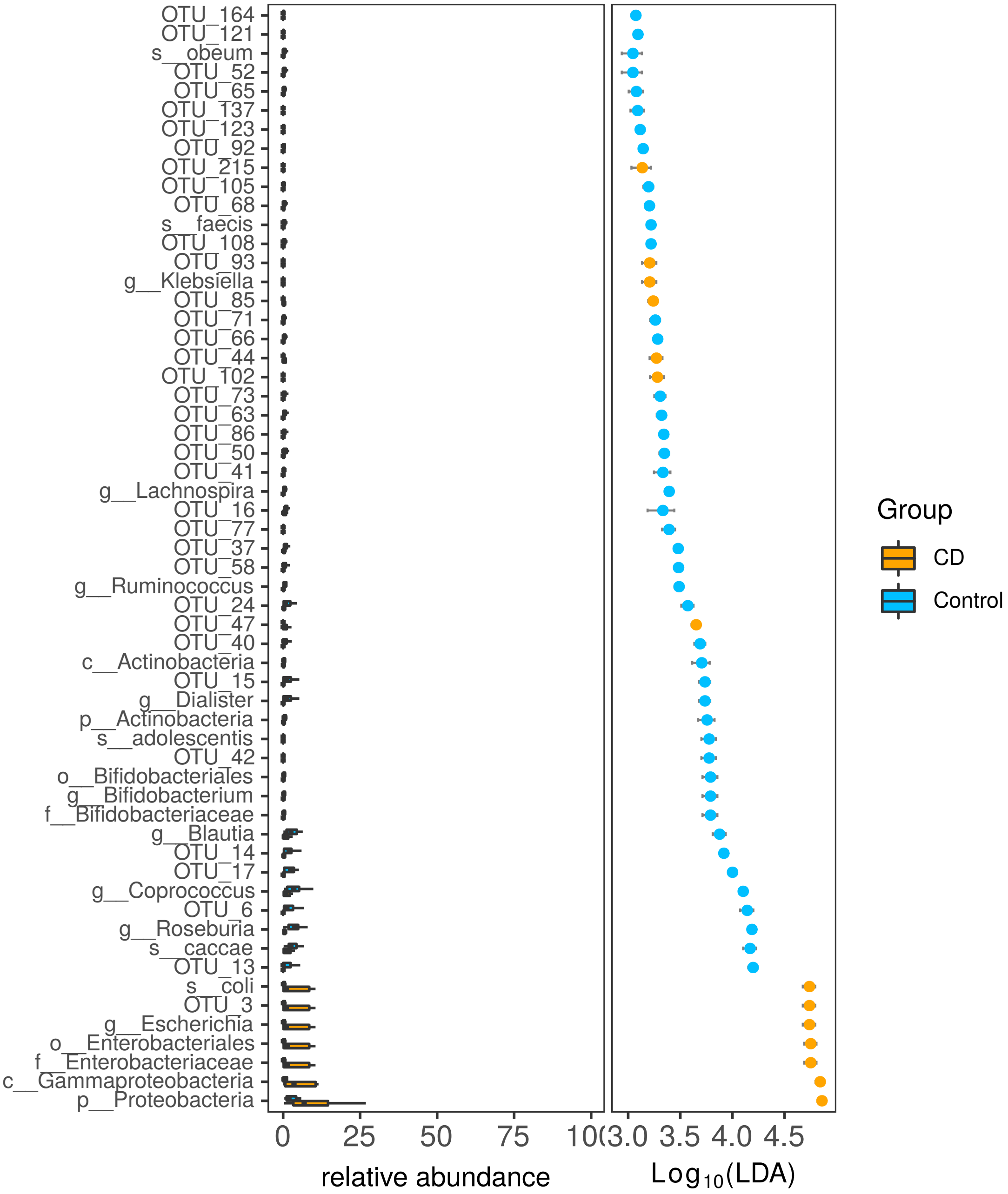


Fig. S13: **The boxplot and the LDA score of different taxa.** The left panel represents the relative abundance of the different taxa, the right panel represents the LDA effect size (95% confidence interval) of different taxa.

1.6.3 visualization of different results by `ggdifftaxbar`

`ggdifftaxbar` can visualize the abundance of biomarker in each samples of groups, the mean and median abundance of groups or subgroups are also showed. `output` parameter is the directory of output.

```r
ggdifftaxbar(obj=deres, xtextsize=1.5,
             output="IBD_biomarkder_barplot",
             coloslist=cols)
```

1.6.4 visualization of different results by `ggeffectsize`

The result is similar with the result of `ggdiffbox`, the bigger confident interval shows that the biomarker is more fluctuant owing to the influence of samples number.

```r
es_p <- ggeffectsize(obj=deres,
                     pointsize = 1,
                     lineheight=0.2,
                     linewidth=0.4) +
        scale_color_manual(values=c(cols)) +
        theme(
          legend.position = "none",
          axis.text = element_text(size = 5),
        )

es_p
```

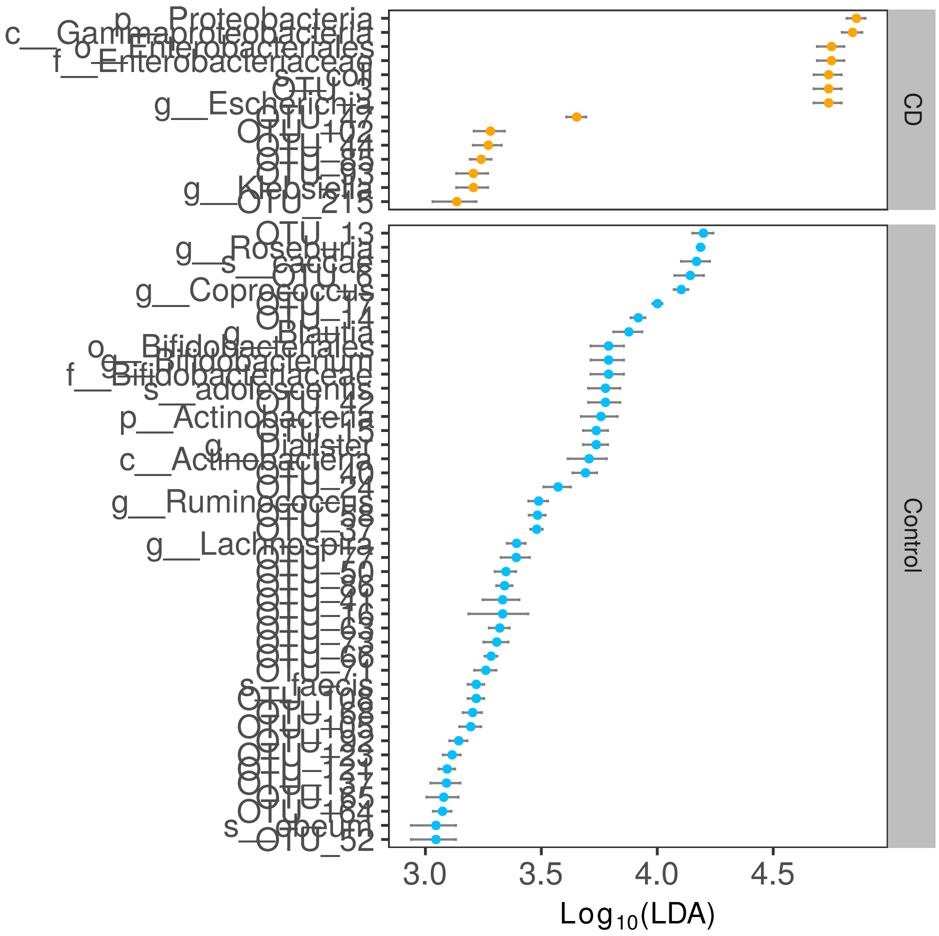


Fig. S14: **The effect size plot of biomarkers**

1.7 Interoperable with the existing computing ecosystem

Because the *MPSE* object *MicrobiotaProcess* inherits the *SummarizedExperiment* object (Morgan et al. 2021), The related inherited methods for signature *SummarizedExperiment* can also be applied to the *MPSE*. For example, the *tidybulk* (Mangiola et al. 2021) provides an R tidy framework for modular transcriptomic data analysis. It provides a *test_differential_abundance* to perform differential transcription testing using edgeR quasi-likelihood edgeR likelihood-ratio (LR), limma-voom, limma-voom-with-quality-weights or DESeq2. It can also be compatible with *MPSE*.

```
library(tidybulk)
library(edgeR)
library(aplot)
library(shadowtext)
library(ggrepel)
mpse2 %<>% test_differential_abundance(.abundance = Abundance, .formula = ~Group)
mpse2
```

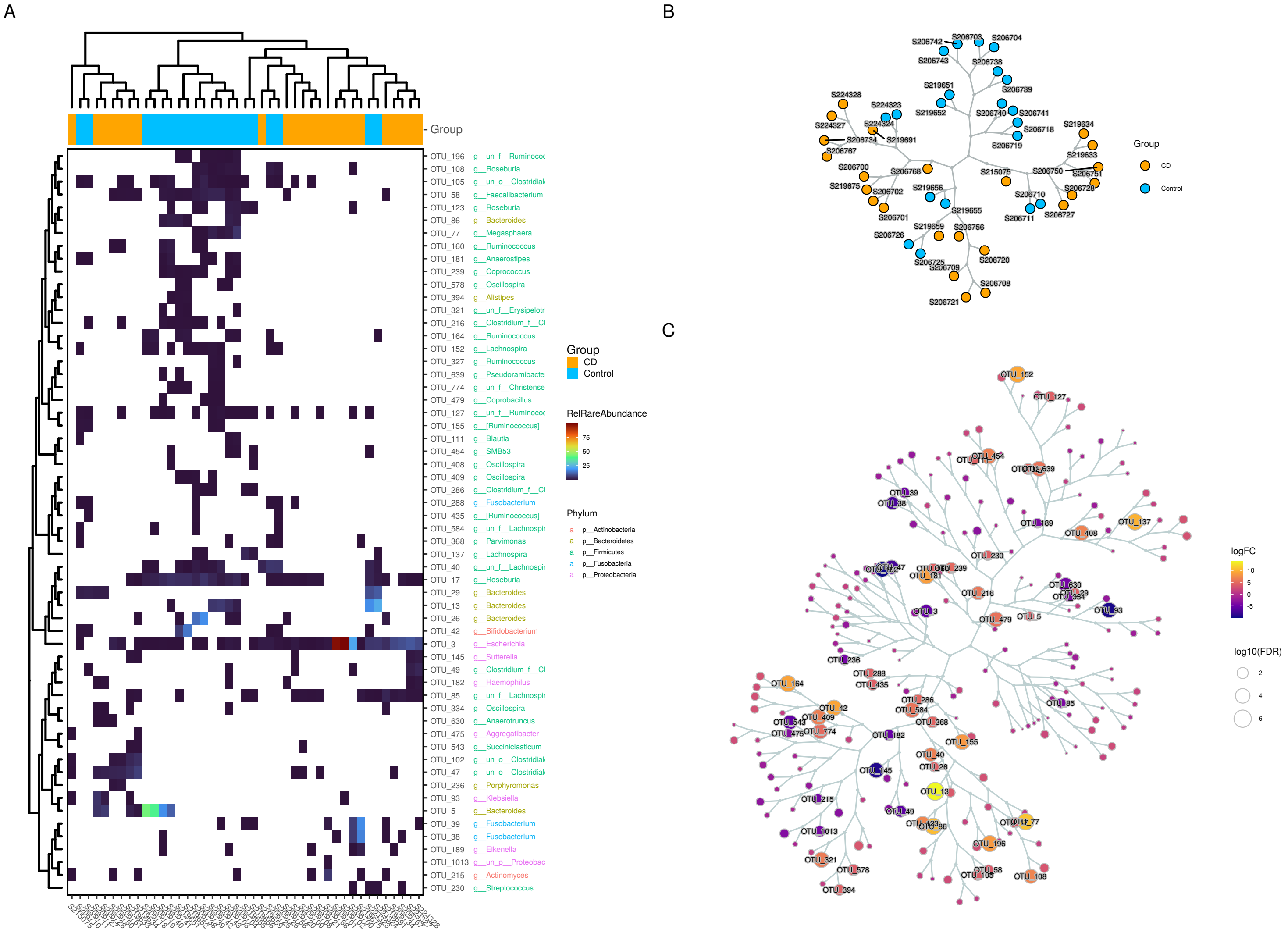
```
## # A MPSE-tibble (MPSE object) abstraction: 9,890 x 41
## # OTU=230 | Samples=43 | Assays=Abundance, RareAbundance, RelRareAbundanceBySample, hellinger | Taxonomy=Kin
##    OTU     Sample  Abundance RareAbundance RelRareAbundanceBySa~ hellinger Group
##    <chr>   <chr>       <int>         <int>                 <dbl>     <dbl> <chr>
##  1 OTU_215 S206700         0             0                     0         0 CD
##  2 OTU_522 S206700         0             0                     0         0 CD
##  3 OTU_719 S206700         0             0                     0         0 CD
##  4 OTU_42  S206700         0             0                     0         0 CD
##  5 OTU_120 S206700         0             0                     0         0 CD
##  6 OTU_138 S206700         0             0                     0         0 CD
##  7 OTU_333 S206700         0             0                     0         0 CD
##  8 OTU_141 S206700         0             0                     0         0 CD
##  9 OTU_322 S206700         0             0                     0         0 CD
## 10 OTU_117 S206700         0             0                     0         0 CD
## # ... with 9,880 more rows, and 34 more variables:
## #   RareAbundanceRarecurve <list>, Observe <dbl>, Chao1 <dbl>, ACE <dbl>,
## #   Shannon <dbl>, Simpson <dbl>, Pielou <dbl>, vennOfGroup <list>,
## #   PC1 (16.56%) <dbl>, PC2 (11.04%) <dbl>, PC3 (9.58%) <dbl>, bray <list>,
## #   PCo1 (19.73%) <dbl>, PCo2 (9.9%) <dbl>, PCo3 (9.11%) <dbl>,
## #   ggupsetOfGroup <list>, logFC <dbl>, logCPM <dbl>, F <dbl>, PValue <dbl>,
## #   FDR <dbl>, Kingdom <chr>, Phylum <chr>, Class <chr>, Order <chr>, ...
```

```
res <- mpse2 %>% dplyr::filter(FDR <= .05 & abs(logFC) >= 2)
pp <- res %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        force = TRUE,
        relative = TRUE,
        feature.dist = "bray",
        geom = "heatmap",
        topn = "all",
        .group = Group
      ) %>%
      set_scale_theme(
        x = list(scale_fill_viridis_c(option = "H"),
                 theme(
                   axis.text.x = element_text(size = 6),
                   axis.text.y = element_text(size = 6),
                   legend.title = element_text(size = 7),
                   legend.text = element_text(size = 5),
                   legend.key.width = unit(0.3, "cm"),
                   legend.key.height = unit(0.3, "cm")
                 )
            ),
        aes_var = RelRareAbundance
      ) %>%
```

```r
      set_scale_theme(
        x = list(scale_fill_manual(values = cols),
                 theme(
                   legend.key.height = unit(0.3, "cm"),
                   legend.key.width = unit(0.3, "cm"),
                   legend.spacing.y = unit(0.02, "cm"),
                   legend.text = element_text(size = 7),
                   legend.title = element_text(size = 9)
                 )
            ),
        aes_var = Group
      )

f <- res %>%
     mp_extract_taxonomy %>%
     ggplot() +
     geom_text(
       mapping = aes(y=OTU, x=0, label=Genus, color=Phylum),
       hjust = 0,
       size = 2
     ) +
     scale_x_continuous(expand=c(0, 0, 0, 0.1)) +
     theme_bw() +
     theme(
       legend.text = element_text(size = 5),
       legend.title = element_text(size = 7),
       legend.key.width = unit(0.3, "cm"),
       legend.key.height = unit(0.3, "cm"),
       panel.background = element_blank(),
       panel.grid = element_blank(),
       axis.text = element_blank(),
       axis.ticks = element_blank(),
       panel.border = element_blank()
     ) +
     labs(x = NULL, y = NULL)

pp <- pp %>% insert_right(f, width = 0.2)

sample.tree <- res %>%
      select(-bray) %>%  # remove the bray, Because it was the result of all OTU,
      mp_cal_clust(.abundance = RelRareAbundanceBySample, distmethod = "bray") %>%
      ggtree(layout = igraph::layout_with_kk, color = "#afb7b8") +
      geom_nodepoint(color = "#afb7b8", size = .5) +
      geom_tippoint(aes(fill = Group), shape = 21, size=3) +
      geom_text_repel(
        data = td_filter(isTip),
        mapping = aes(label = label),
        color = "black",
        bg.color = "grey",
        size = 2,
        max.overlaps = 30
      ) +
      scale_fill_manual(
        values = cols,
        guide = guide_legend(
           title.theme = element_text(size = 7),
           label.theme = element_text(size = 5),
        )
      )
```

```
## # Note: MPSE object is converted to a tibble data (tbl_mpse object) for independent data analysis.
## #    A new MPSE object can be returned by setting keep.mpse = TRUE.
```

```
p <- mpse2 %>%
      mp_cal_dist(
         .abundance = RelRareAbundanceBySample,
         distmethod = "bray",
         cal.feature.dist = T
      ) %>%
      hclust() %>%
      ggtree(layout = igraph::layout_with_kk, color = "#bed0d1") +
      geom_nodepoint(color = "#bed0d1", size = .5)

# The data.frame contained results of test_differential_abundance
otu.tab <- mpse2 %>% mp_extract_feature()
p <- p %<+% otu.tab +
     geom_tippoint(
       mapping = aes(fill = logFC, size = -log10(FDR)),
       shape = 21,
       color = "grey"
     ) +
     scale_fill_viridis_c(
       option="C",
       guide = guide_colorbar(
          title.theme = element_text(size = 7),
          label.theme = element_text(size = 5),
          barheight = unit(1.5, "cm"),
          barwidth = unit(.3, "cm")
       )
     ) +
     scale_size_continuous(
       range = c(.5, 6),
       guide = guide_legend(
          key.width = .3,
          key.height = .3,
          label.theme = element_text(size = 5),
          title.theme = element_text(size = 7)
       )
     )+
     geom_shadowtext(
       data = td_filter(FDR <= .05 & abs(logFC) >= 2),
       mapping = aes(x = x, y = y, label = label),
       color = "black",
       bg.color = "grey",
       size = 2
       #max.overlaps = 60,
     )

design <- "
  12
  13
  13
"

plot_list(pp, sample.tree, p, design = design, tag_levels = "A")
```


A
B
C
Group
OTU_196 g__un_f__Ruminococ
OTU_108 g__Roseburia
OTU_105 g__un_o__Clostridial
OTU_58 g__Faecalibacterium
OTU_123 g__Roseburia
OTU_86 g__Bacteroides
OTU_77 g__Megasphaera
OTU_160 g__Ruminococcus
OTU_181 g__Anaerostipes
OTU_239 g__Coprococcus
OTU_578 g__Oscillospira
OTU_394 g__Alistipes
OTU_321 g__un_f__Erysipelotr
OTU_216 g__Clostridium_f__Cl
OTU_164 g__Ruminococcus
OTU_152 g__Lachnospira
OTU_327 g__Ruminococcus
OTU_639 g__Pseudoramibacte
OTU_774 g__un_f__Christense
OTU_479 g__Coprobacillus
OTU_127 g__un_f__Ruminococ
OTU_155 g__[Ruminococcus]
OTU_111 g__Blautia
OTU_454 g__SMB53
OTU_408 g__Oscillospira
OTU_409 g__Oscillospira
OTU_286 g__Clostridium_f__Cl
OTU_288 g__Fusobacterium
OTU_435 g__[Ruminococcus]
OTU_584 g__un_f__Lachnospi
OTU_368 g__Parvimonas
OTU_137 g__Lachnospira
OTU_40 g__un_f__Lachnospi
OTU_17 g__Roseburia
OTU_29 g__Bacteroides
OTU_13 g__Bacteroides
OTU_26 g__Bacteroides
OTU_42 g__Bifidobacterium
OTU_3 g__Escherichia
OTU_145 g__Sutterella
OTU_49 g__Clostridium_f__Cl
OTU_182 g__Haemophilus
OTU_85 g__un_f__Lachnospi
OTU_334 g__Oscillospira
OTU_630 g__Anaerotruncus
OTU_475 g__Aggregatibacter
OTU_543 g__Succiniclasticum
OTU_102 g__un_o__Clostridial
OTU_47 g__un_o__Clostridial
OTU_236 g__Porphyromonas
OTU_93 g__Klebsiella
OTU_5 g__Bacteroides
OTU_39 g__Fusobacterium
OTU_38 g__Fusobacterium
OTU_189 g__Eikenella
OTU_1013 g__un_p__Proteobac
OTU_215 g__Actinomyces
OTU_230 g__Streptococcus
Group
CD
Control
RelRareAbundance
75
50
25
Phylum
a p__Actinobacteria
a p__Bacteroidetes
a p__Firmicutes
a p__Fusobacteria
a p__Proteobacteria
S206742
S206703
S206704
S206743
S206738
S219651
S206739
S224328
S224323
S219652
S206740
S206741
S224327
S224324
S206718
S219634
S206734
S219691
S206719
S206767
S219633
S206700
S206768
S215075
S206750
S206751
S219675
S219656
S206728
S206702
S206710
S219655
S206701
S206711
S206727
S219659
S206756
S206726
S206725
S206709
S206720
S206708
S206721
Group
CD
Control
OTU_152
OTU_127
OTU_454
OTU_111
OTU_327
OTU_639
OTU_39
OTU_38
OTU_189
OTU_137
OTU_408
OTU_230
OTU_102
OTU_47
OTU_160
OTU_239
OTU_181
OTU_216
OTU_630
OTU_29
OTU_334
OTU_13
OTU_93
OTU_479
OTU_5
OTU_236
OTU_288
OTU_435
OTU_164
OTU_286
OTU_584
OTU_42
OTU_409
OTU_543
OTU_475
OTU_774
OTU_368
OTU_182
OTU_155
OTU_40
OTU_26
OTU_145
OTU_13
OTU_215
OTU_49
OTU_123
OTU_86
OTU_17
OTU_77
OTU_1013
OTU_196
OTU_321
OTU_578
OTU_58
OTU_105
OTU_108
OTU_394
logFC
10
5
0
-5
-log10(FDR)
2
4
6

2 the analysis of the other published pediatric CD stool samples

In the previous chapter, we described how to use *MicrobiotaProcess* to do the analysis of the 16s rDNA data. However, it also can be applied to metagenome or metatranscriptome species community data and the function data analysis. In this chapter, we used the example datasets about the other published pediatric CD stool microbial study [@] to show how to used *MicrobiotaProcess* to do the related analysis. The datasets were obtained from the github[2].

2.1 Loading the 16s data and construction of MPSE class

The chapter is similar with the 1, so some operations can refer to the previous chapter 1.

```
cols <- c("orange", "deepskyblue")
cols2 <- c("deepskyblue", "yellow", "#FF9933")
sample.da <- read.table("./data/CD_RF_microbiome/biscuit_metadata.txt", header=TRUE, check.names=FALSE, sep="\t
sample.da %<>% dplyr::select(1:5)
biom <- biomformat::read_biom("./data/CD_RF_microbiome/otu_table_w_tax_BISCUIT.biom")
mpse16s <- biom %>% as.MPSE
mpse16s
```

```
## # A MPSE-tibble (MPSE object) abstraction: 37,392 x 10
## # OTU=984 | Samples=38 | Assays=Abundance | Taxanomy=Kingdom, Phylum, Class, Order, Family, Genus, Speies
##    OTU     Sample Abundance Kingdom     Phylum  Class  Order Family Genus Speies
##    <chr>   <chr>      <dbl> <chr>       <chr>   <chr>  <chr> <chr>  <chr> <chr>
##  1 358030  S15            5 k__Bacteria p__Fir~ c__Cl~ o__C~ f__Ru~ g__u~ s__un~
##  2 196271  S15            0 k__Bacteria p__Fir~ c__Cl~ o__C~ f__La~ g__u~ s__un~
##  3 196270  S15            2 k__Bacteria p__Fir~ c__Cl~ o__C~ f__un~ g__u~ s__un~
##  4 297149  S15            0 k__Bacteria p__Fir~ c__Cl~ o__C~ f__La~ g__u~ s__un~
##  5 3604981 S15            0 k__Bacteria p__Fir~ c__Cl~ o__C~ f__La~ g__B~ s__un~
##  6 240755  S15            0 k__Bacteria p__Pro~ c__Ga~ o__P~ f__Pa~ g__H~ s__in~
##  7 326482  S15            0 k__Bacteria p__Bac~ c__Ba~ o__B~ f__Pr~ g__P~ s__co~
##  8 4393540 S15            0 k__Bacteria p__Bac~ c__Ba~ o__B~ f__[B~ g__u~ s__un~
##  9 4339144 S15            0 k__Bacteria p__Bac~ c__Ba~ o__B~ f__[O~ g__B~ s__un~
## 10 4369050 S15            0 k__Bacteria p__Fus~ c__Fu~ o__F~ f__Fu~ g__F~ s__un~
## # ... with 37,382 more rows
```

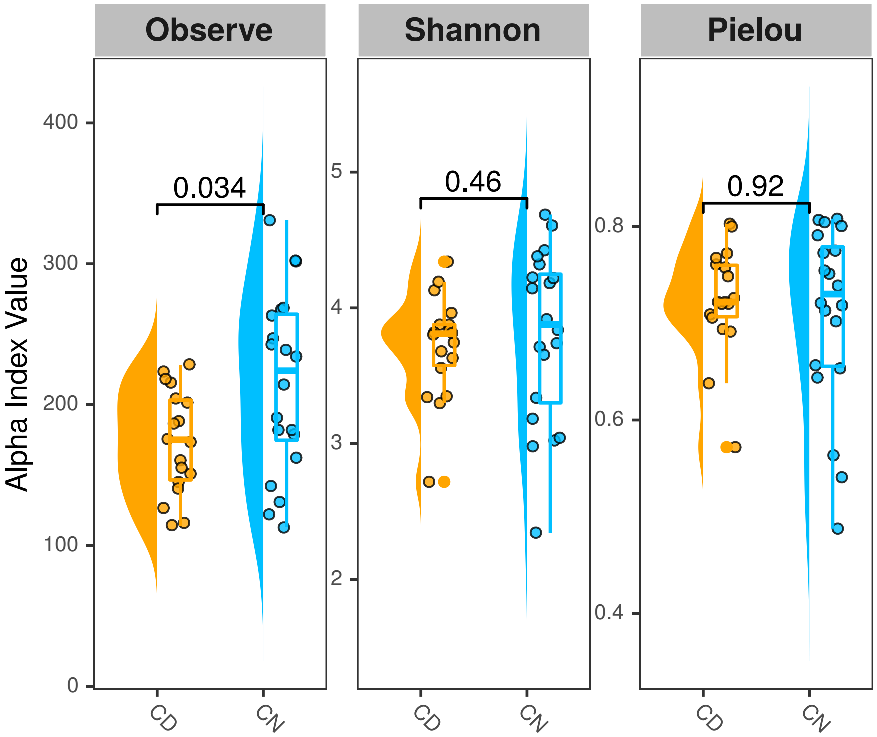
```
mpse16s %<>% dplyr::left_join(sample.da, by=c("Sample"="sample_id"))
mpse16s
```

```
## # A MPSE-tibble (MPSE object) abstraction: 37,392 x 14
## # OTU=984 | Samples=38 | Assays=Abundance | Taxanomy=Kingdom, Phylum, Class, Order, Family, Genus, Speies
##    OTU     Sample Abundance disease response sex     age Kingdom  Phylum  Class
##    <chr>   <chr>      <dbl> <chr>   <chr>    <chr> <dbl> <chr>    <chr>   <chr>
##  1 358030  S15            5 CN      CN       Male   15.4 k__Bact~ p__Fir~ c__Cl~
##  2 196271  S15            0 CN      CN       Male   15.4 k__Bact~ p__Fir~ c__Cl~
##  3 196270  S15            2 CN      CN       Male   15.4 k__Bact~ p__Fir~ c__Cl~
##  4 297149  S15            0 CN      CN       Male   15.4 k__Bact~ p__Fir~ c__Cl~
##  5 3604981 S15            0 CN      CN       Male   15.4 k__Bact~ p__Fir~ c__Cl~
##  6 240755  S15            0 CN      CN       Male   15.4 k__Bact~ p__Pro~ c__Ga~
##  7 326482  S15            0 CN      CN       Male   15.4 k__Bact~ p__Bac~ c__Ba~
##  8 4393540 S15            0 CN      CN       Male   15.4 k__Bact~ p__Bac~ c__Ba~
##  9 4339144 S15            0 CN      CN       Male   15.4 k__Bact~ p__Bac~ c__Ba~
## 10 4369050 S15            0 CN      CN       Male   15.4 k__Bact~ p__Fus~ c__Fu~
## # ... with 37,382 more rows, and 4 more variables: Order <chr>, Family <chr>,
## #   Genus <chr>, Speies <chr>
```

2.1.1 Alpha diversity analysis in 16s level

[2] https://github.com/LangilleLab/CD__RF__microbiome

```
mpse16s %<>%
    mp_rrarefy() %>%
    mp_cal_alpha(.abundance = RareAbundance)
p <- mpse16s %>%
     mp_plot_alpha(
        .group = disease,
        .alpha = c(Observe, Shannon, Pielou)
     ) +
     scale_fill_manual(values = cols) +
     scale_color_manual(values = cols) +
     theme(legend.position = "none")
p
```



2.1.2 Beta diversity analysis in 16s level

```
mpse16s %<>%
    mp_decostand(
      .abundance = Abundance,
      method = "hellinger"
    )

mpse16s %<>%
    mp_cal_dist(
      .abundance = hellinger,
      distmethod = "bray"
    )

mpse16s %<>%
    mp_cal_pcoa(
      .abundance = hellinger,
      distmethod = 'bray'
    )

p1 <- mpse16s %>%
      mp_plot_dist(
        .distmethod = bray,
        .group = c(disease, response)
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(
              values = cols,
              guide = guide_legend(
                         keywidth = 0.5, keyheight = 0.5,
                         label.theme=element_text(size=6)
                )
            ),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(
              values=cols2,
              guide = guide_legend(
                         keywidth = 0.5,
                         keyheight = 0.5,
                         label.theme=element_text(size=6)
                )
            ),
        aes_var = response
      ) %>%
      set_scale_theme(
        x = scale_size_continuous(
              range = c(1, 3),
              guide = guide_legend(
                         keywidth = 0.5,
                         keyheight = 0.5,
                         label.theme=element_text(size=6)
              )
            ),
        aes_var = bray
      )

p2 <- mpse16s %>%
      mp_plot_dist(
        .distmethod = bray,
        .group = disease,
        group.test = TRUE
      ) +
      scale_color_manual(
        values = c("orange", "#00A08A", "deepskyblue")
      ) +
      scale_fill_manual(
        values = c("orange", "#00A08A", "deepskyblue")
      )

p3 <- mpse16s %>%
      mp_plot_ord(
        .ord = pcoa,
        .group = disease,
        .size = Observe,
        .starshape = response,
        show.side = FALSE
      ) +
      scale_starshape_manual(values = c(1, 13, 15)) +
      scale_fill_manual(
        values = cols,
        guide = guide_legend(override.aes=list(size=2, starshape = 15))
      ) +
      scale_size_continuous(
        range = c(1, 3),
        guide = guide_legend(override.aes=list(starshape = 15))
      ) +
      theme(
        legend.key.height = unit(0.3, "cm"),
        legend.key.width = unit(0.3, "cm"),
        legend.spacing.y = unit(0.02, "cm"),
        legend.text = element_text(size = 7),
        legend.title = element_text(size = 9),
      )

ff <- aplot::plot_list(p1, (aplot::plot_list(p2, p3, nrow=1, widths=c(1, 2))), ncol = 1, heights = c(1.4, 1))
ff
```

```
mpse16s %>%
    mp_adonis(
       .abundance = Abundance,
       .formula = ~ disease + response,
       distmethod = "bray",
       permutation = 9999
     )
```

```
##
## Call:
## vegan::adonis(formula = .formula, data = sampleda, permutations = permutations,      method = distmethod)
##
## Permutation: free
## Number of permutations: 9999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## disease    1    0.4244 0.42438 1.25679 0.03390 0.1462
## response   1    0.2760 0.27600 0.81737 0.02205 0.7677
## Residuals 35   11.8185 0.33767         0.94405
## Total     37   12.5189                 1.00000
```

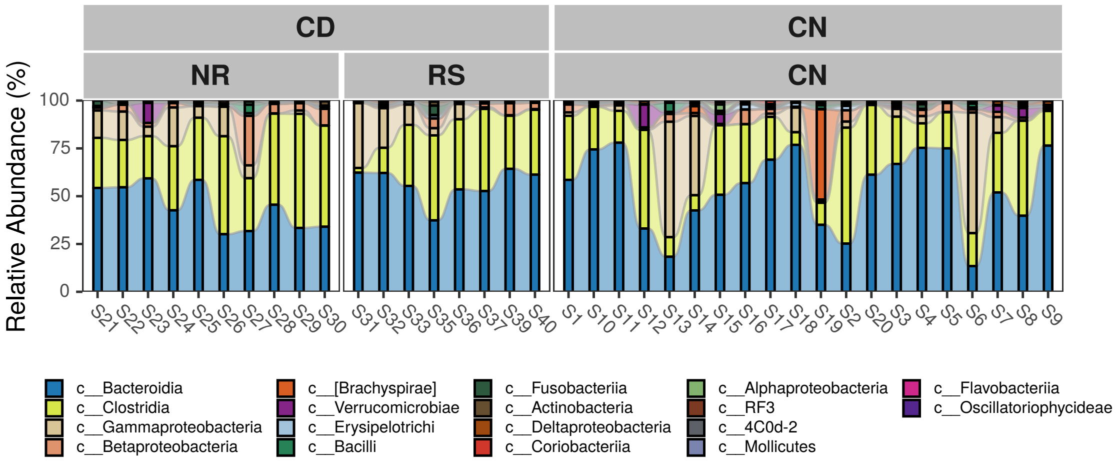
2.1.3 Composition of the taxonomy in 16s level

```
mpse16s %<>%
    mp_cal_abundance(
       .abundance=RareAbundance
    )
```

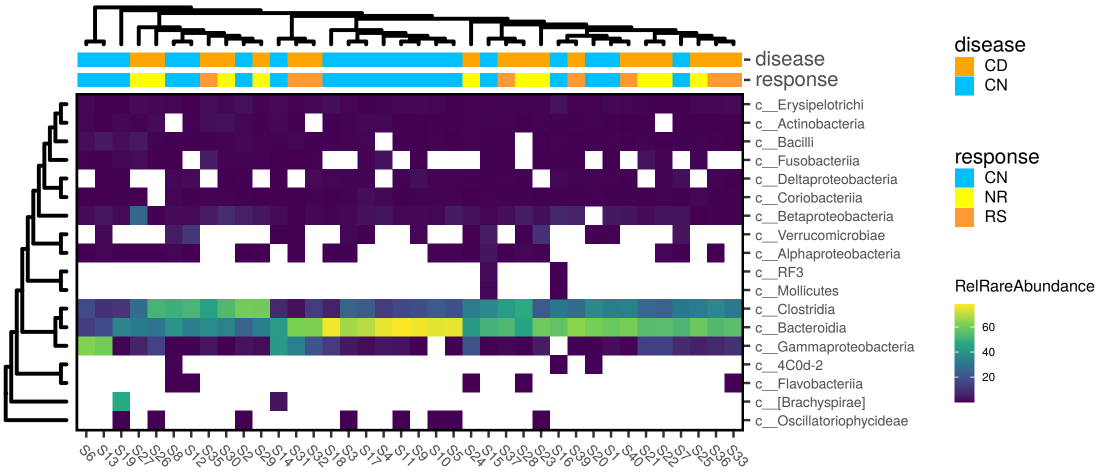
```
## The otutree is empty in the MPSE object!
```

```
p1 <- mpse16s %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        taxa.class = Class,
        topn = 20,
        .group = c(disease, response)
      ) +
      theme(
        legend.key.height = unit(0.3, "cm"),
        legend.key.width = unit(0.3, "cm"),
        legend.spacing.y = unit(0.02, "cm"),
        legend.text = element_text(size = 7)
      )

p2 <- mpse16s %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        taxa.class = Class,
        topn = 20,
        .group = c(disease, response)
      ) +
      theme(
        legend.key.height = unit(0.3, "cm"),
        legend.key.width = unit(0.3, "cm"),
        legend.spacing.y = unit(0.02, "cm"),
        legend.text = element_text(size = 7)
      )

p3 <- mpse16s %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        taxa.class = Class,
        topn = 20,
        .group = c(response, disease),
        geom = "heatmap"
      ) %>%
      set_scale_theme(
        x = scale_fill_viridis_c(),
        aes_var = RelRareAbundance
      ) %>%
      set_scale_theme(
        x = theme(
              axis.text = element_text(size = 6),
              legend.title = element_text(size = 7),
              legend.text = element_text(size=5),
              legend.key.width = unit(0.3, "cm"),
              legend.key.height=unit(0.3, "cm")
            ),
        aes_var = RelRareAbundance
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(values = cols),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = theme(
          legend.key.height = unit(0.3, "cm"),
          legend.key.width = unit(0.3, "cm"),
          legend.spacing.y = unit(0.02, "cm"),
          legend.text = element_text(size = 7),
          legend.title = element_text(size = 9),
         ),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(values = cols2),
        aes_var = response
      ) %>%
      set_scale_theme(
         x = theme(
          legend.key.height = unit(0.3, "cm"),
          legend.key.width = unit(0.3, "cm"),
          legend.spacing.y = unit(0.02, "cm"),
          legend.text = element_text(size = 7),
          legend.title = element_text(size = 9),
         ),
        aes_var = response
      )

p4 <- mpse16s %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        taxa.class = Class,
        topn = 20,
        .group = c(response, disease),
        geom = "heatmap"
      ) %>%
      set_scale_theme(
        x = scale_fill_viridis_c(),
        aes_var = RelRareAbundance
      ) %>%
      set_scale_theme(
        x = theme(
              axis.text = element_text(size = 6),
              legend.title = element_text(size = 7),
              legend.text = element_text(size = 5),
              legend.key.width = unit(0.3, "cm"),
              legend.key.height=unit(0.3, "cm")
            ),
        aes_var = RelRareAbundance
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(values = cols),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = theme(
          legend.key.height = unit(0.3, "cm"),
          legend.key.width = unit(0.3, "cm"),
          legend.spacing.y = unit(0.02, "cm"),
          legend.text = element_text(size = 7),
          legend.title = element_text(size = 9),
         ),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(values = cols2),
        aes_var = response
      ) %>%
      set_scale_theme(
          x = theme(
            legend.key.height = unit(0.3, "cm"),
            legend.key.width = unit(0.3, "cm"),
            legend.spacing.y = unit(0.02, "cm"),
            legend.text = element_text(size = 7),
            legend.title = element_text(size = 9),
           ),
          aes_var = response
        )
```

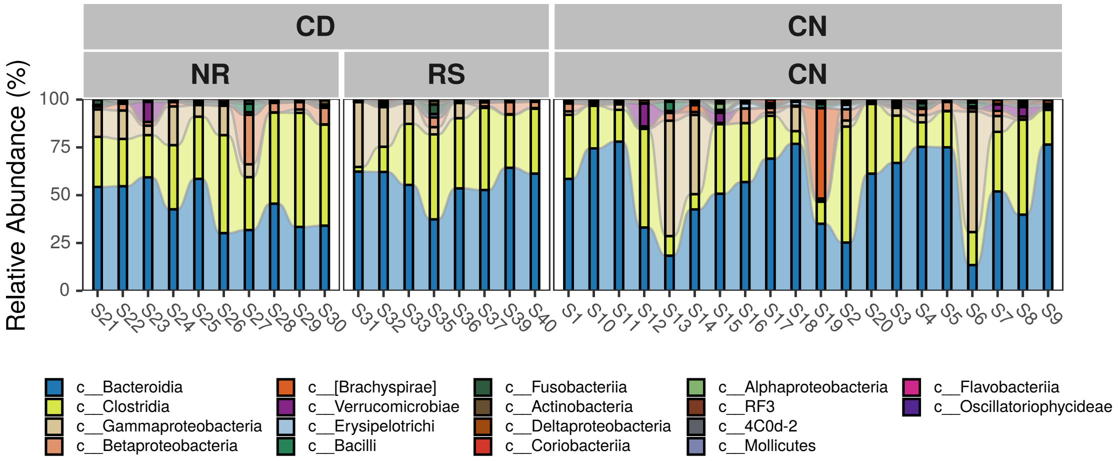
```
aplot::plot_list(p1, p3, p2, p4, ncol = 1)
```



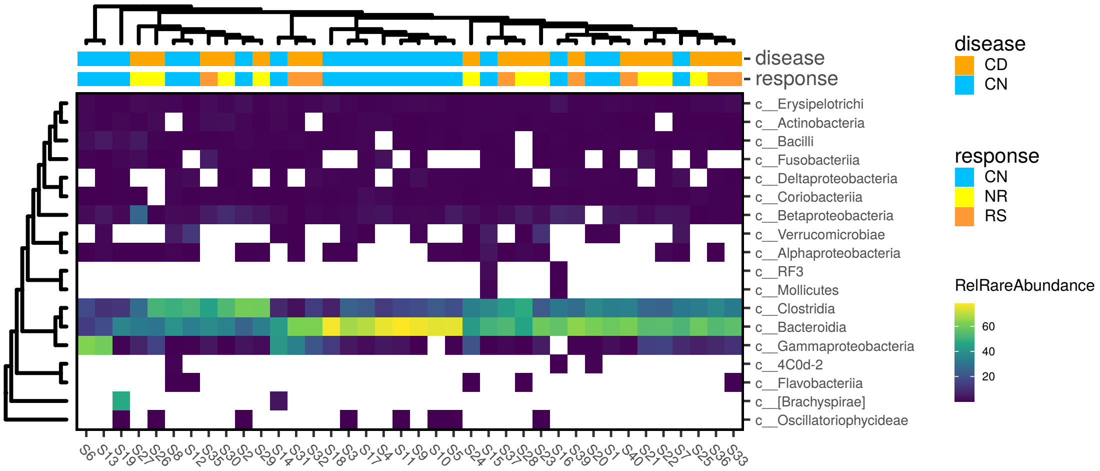
CD
CN
NR
RS
CN
Relative Abundance (%)
100
75
50
25
0
S21 S22 S23 S24 S25 S26 S27 S28 S29 S30
S31 S32 S33 S35 S36 S37 S39 S40
S1 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S2 S20 S3 S4 S5 S6 S7 S8 S9
c__Bacteroidia
c__Clostridia
c__Gammaproteobacteria
c__Betaproteobacteria
c__[Brachyspirae]
c__Verrucomicrobiae
c__Erysipelotrichi
c__Bacilli
c__Fusobacteriia
c__Actinobacteria
c__Deltaproteobacteria
c__Coriobacteriia
c__Alphaproteobacteria
c__RF3
c__4C0d-2
c__Mollicutes
c__Flavobacteriia
c__Oscillatoriophycideae



disease
response
c__Erysipelotrichi
c__Actinobacteria
c__Bacilli
c__Fusobacteriia
c__Deltaproteobacteria
c__Coriobacteriia
c__Betaproteobacteria
c__Verrucomicrobiae
c__Alphaproteobacteria
c__RF3
c__Mollicutes
c__Clostridia
c__Bacteroidia
c__Gammaproteobacteria
c__4C0d-2
c__Flavobacteriia
c__[Brachyspirae]
c__Oscillatoriophycideae
S6 S13 S19 S27 S26 S8 S12 S25 S30 S2 S29 S14 S31 S32 S18 S3 S17 S4 S11 S9 S10 S5 S24 S15 S37 S28 S23 S16 S39 S20 S1 S40 S21 S22 S7 S35 S36 S33
disease
CD
CN
response
CN
NR
RS
RelRareAbundance
60
40
20



CD
CN
NR
RS
CN
Relative Abundance (%)
100
75
50
25
0
S21 S22 S23 S24 S25 S26 S27 S28 S29 S30
S31 S32 S33 S35 S36 S37 S39 S40
S1 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S2 S20 S3 S4 S5 S6 S7 S8 S9
c__Bacteroidia
c__Clostridia
c__Gammaproteobacteria
c__Betaproteobacteria
c__[Brachyspirae]
c__Verrucomicrobiae
c__Erysipelotrichi
c__Bacilli
c__Fusobacteriia
c__Actinobacteria
c__Deltaproteobacteria
c__Coriobacteriia
c__Alphaproteobacteria
c__RF3
c__4C0d-2
c__Mollicutes
c__Flavobacteriia
c__Oscillatoriophycideae

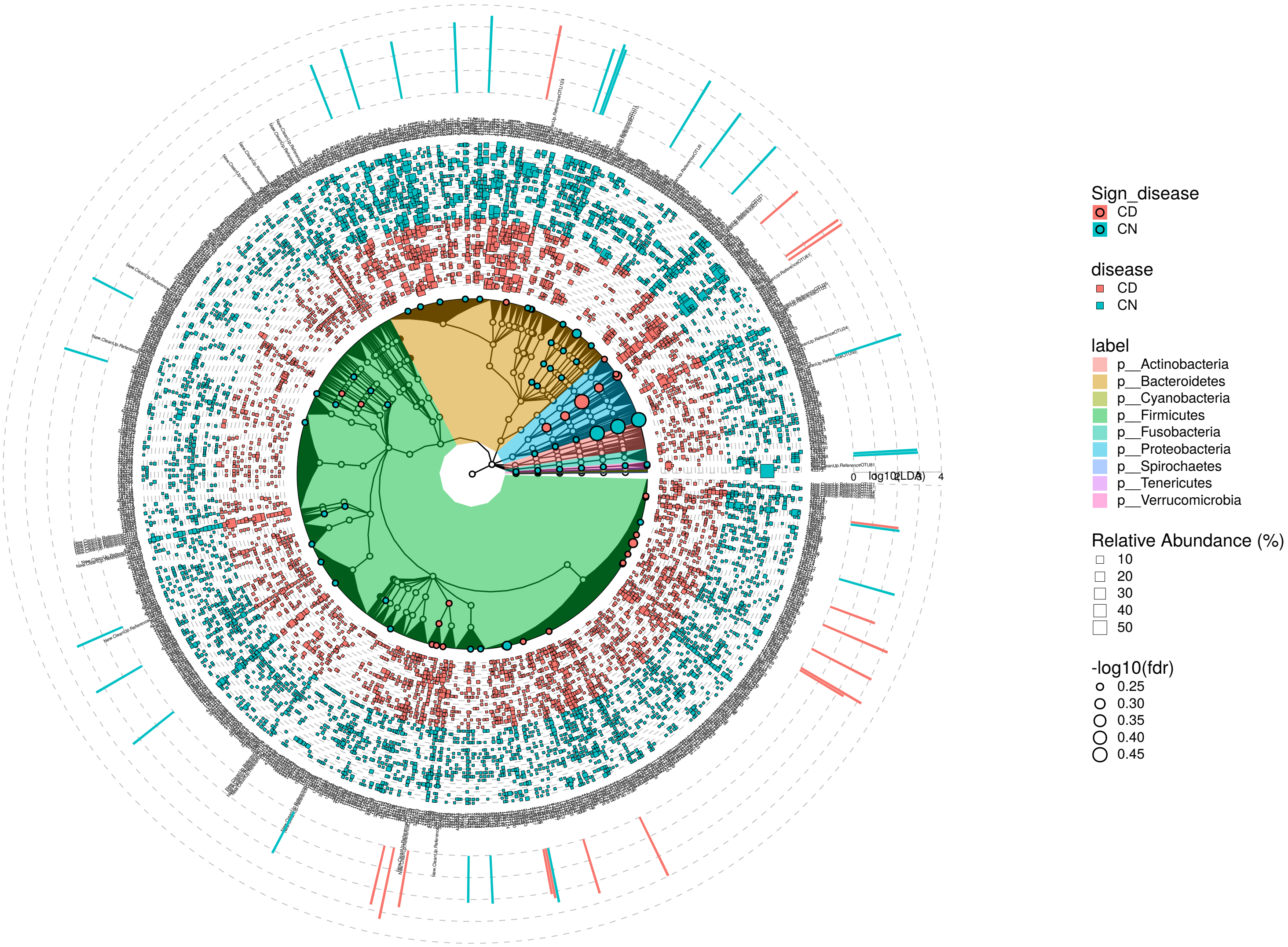


disease
response
c__Erysipelotrichi
c__Actinobacteria
c__Bacilli
c__Fusobacteriia
c__Deltaproteobacteria
c__Coriobacteriia
c__Betaproteobacteria
c__Verrucomicrobiae
c__Alphaproteobacteria
c__RF3
c__Mollicutes
c__Clostridia
c__Bacteroidia
c__Gammaproteobacteria
c__4C0d-2
c__Flavobacteriia
c__[Brachyspirae]
c__Oscillatoriophycideae
S6 S13 S19 S27 S26 S8 S12 S25 S30 S2 S29 S14 S31 S32 S18 S3 S17 S4 S11 S9 S10 S5 S24 S15 S37 S28 S23 S16 S39 S20 S1 S40 S21 S22 S7 S35 S36 S33
disease
CD
CN
response
CN
NR
RS
RelRareAbundance
60
40
20

2.1.4 Different analysis in 16S level

MicrobiotaProcess also provides *mp_plot_diff_res* to displayed the result of *mp_diff_analysis* with *action="add"*, which can decreases coding burden.

```
mpse16s %<>%
    mp_diff_analysis(.abundance = RareAbundance, .group = disease, filter.p = "pvalue")

p <- mpse16s %>%
     mp_plot_diff_res(tiplab.size = 0.8)
p
```



2.2 Loading the KEGG three levels data

The KEGG pathway abundances were predicted based on the 16s rDNA data. It can also be imported as MPSE, and further analyzed using *MicrobiotaProcess*. Here, we only show how to identify the different pathway using the *mp_diff_analysis* of *MicrobiotaProcess* (refer to 2.3). Other operations are similar with the analysis of 16s rDNA data (refer to 1).

```
xx <- read.table("./data/CD_RF_microbiome/ko_l3.spf", header=TRUE, sep="\t", check.names=FALSE, comment.char=""
taxa.da <- xx %>%
           select(Level_1, Level_2, Level_3) %>%
           tibble::column_to_rownames(var="Level_3") %>%
           convert_to_treedata(type="others", include.rownames=TRUE)
abun.da <- xx %>%
           select(-c(Level_1, Level_2)) %>%
          tibble::column_to_rownames(var="Level_3")

mpseKEGG <- MPSE(assays = abun.da, taxatree = taxa.da)
mpseKEGG %<>% dplyr::left_join(sample.da, by = c("Sample" = "sample_id"))
mpseKEGG %<>% mp_rrarefy()
mpseKEGG
```

```
## # A MPSE-tibble (MPSE object) abstraction: 12,464 x 10
## # OTU=328 | Samples=38 | Assays=Abundance, RareAbundance | Taxanomy=Level_1, Level_2
##    OTU      Sample Abundance RareAbundance disease response sex     age Level_1
##    <chr>    <chr>      <int>         <int> <chr>   <chr>    <chr> <dbl> <chr>
##  1 1,1,1-T~ S15            6             4 CN      CN       Male   15.4 d1__Met~
##  2 ABC tra~ S15        75667         58385 CN      CN       Male   15.4 d1__Env~
##  3 Adheren~ S15            0             0 CN      CN       Male   15.4 d1__Cel~
##  4 Adipocy~ S15         2506          1913 CN      CN       Male   15.4 d1__Org~
##  5 African~ S15          118            90 CN      CN       Male   15.4 d1__Hum~
##  6 Alanine~ S15        31759         24627 CN      CN       Male   15.4 d1__Met~
##  7 Aldoste~ S15            0             0 CN      CN       Male   15.4 d1__Org~
##  8 Alzheim~ S15         1985          1527 CN      CN       Male   15.4 d1__Hum~
##  9 Amino a~ S15         5560          4288 CN      CN       Male   15.4 d1__Met~
## 10 Amino a~ S15        42320         32676 CN      CN       Male   15.4 d1__Met~
## # ... with 12,454 more rows, and 1 more variable: Level_2 <chr>
```

2.2.1 Different analysis in KEGG level

```
mpseKEGG %<>%
   mp_diff_analysis(
     .abundance = RareAbundance,
     .group = disease,
     filter.p = "pvalue"

   )
```

```
## The otutree is empty in the MPSE object!
```

```
mpseKEGG %>% mp_plot_diff_res(.taxa.class = Level_1)
```

```
mpseKEGG %>% mp_extract_tree() %>% dplyr::filter(!is.na(pvalue) & pvalue <=0.05, keep.td=FALSE)
```

```
## # A tibble: 7 x 12
##    node label        isTip nodeClass nodeDepth RareAbundanceBy~ LDAupper LDAmean
##   <dbl> <chr>        <lgl> <chr>         <dbl> <list>              <dbl>   <dbl>
## 1    75 Cytoskeleto~ TRUE  OTU               3 <tibble [38 x 6~     2.29    2.24
## 2    78 Cell motili~ TRUE  OTU               3 <tibble [38 x 6~     2.14    2.09
## 3   105 Photosynthe~ TRUE  OTU               3 <tibble [38 x 6~     2.32    2.27
## 4   203 Porphyrin a~ TRUE  OTU               3 <tibble [38 x 6~     2.80    2.75
## 5   250 Protein fol~ TRUE  OTU               3 <tibble [38 x 6~     2.29    2.24
## 6   350 d2__Cell mo~ FALSE Level_2           2 <tibble [38 x 6~     2.14    2.09
## 7   386 d2__Protein~ FALSE Level_2           2 <tibble [38 x 6~     2.29    2.24
## # ... with 4 more variables: LDAlower <dbl>, Sign_disease <chr>, pvalue <dbl>,
## #   fdr <dbl>
```

2.3 Loading the MGS data

The taxa abundance data also can be analyzed by *MicrobiotaProcess*, Here we used the example data from output of *MetaPhlAn* (Segata et al. 2012) to show how to perform the related analysis using *MicrobiotaProcess*. The output of other taxa assign and qu can also be imported and converted to the *MPSE* object, and be further analyzed using *MicrobiotaProcess*, which can refer to 2.2 and 3

```
mpseMGS <- mp_import_metaphlan("./data/CD_RF_microbiome/metaphlan2_out_merged_species.tsv", linenum=1)
colnames(mpseMGS) <- mpseMGS %>% mp_extract_sample %>% pull(2)
mpseMGS %<>% left_join(sample.da, by=c("Sample"="sample_id"))
mpseMGS
```

```
## # A MPSE-tibble (MPSE object) abstraction: 4,370 x 14
## # OTU=115 | Samples=38 | Assays=Abundance | Taxanomy=Kingdom, Phylum, Class, Order, Family, Genus
##    OTU     Sample Abundance unknown1 disease response sex     age Kingdom Phylum
##    <chr>   <chr>      <dbl> <chr>    <chr>   <chr>    <chr> <dbl> <chr>   <chr>
##  1 s__un_~ S12         0    S12      CN      CN       Fema~   8.6 k__Arc~ p__Eu~
##  2 s__Bif~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  3 s__Bif~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  4 s__Bif~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  5 s__Col~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  6 s__Col~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  7 s__un_~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  8 s__un_~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ac~
##  9 s__Bac~ S12         6.34 S12      CN      CN       Fema~   8.6 k__Bac~ p__Ba~
## 10 s__Bac~ S12         0    S12      CN      CN       Fema~   8.6 k__Bac~ p__Ba~
## # ... with 4,360 more rows, and 4 more variables: Class <chr>, Order <chr>,
## #   Family <chr>, Genus <chr>
```

2.3.1 Alpha diversity analysis in MGS level

```
mpseMGS %<>%
    mp_cal_alpha(
      .abundance = Abundance,
      force = TRUE
    )
p <- mpseMGS %>%
     mp_plot_alpha(
       .group = disease,
       .alpha = c(Observe, Shannon, Pielou)
     ) +
     scale_color_manual(values = cols) +
     scale_fill_manual(values = cols) +
     theme(legend.position = "none")
p
```

2.3.2 Beta diversity analysis in MGS level

```
mpseMGS %<>%
   mp_decostand(
      .abundance = Abundance,
      method = "hellinger"
    )

mpseMGS %<>%
    mp_cal_dist(
      .abundance = hellinger,
      distmethod = "bray"
    )

mpseMGS %<>%
    mp_cal_pcoa(
      .abundance = hellinger,
      distmethod = "bray"
    )
```

```
p1 <- mpseMGS %>%
      mp_plot_dist(
        .distmethod = bray,
        .group=c(disease, response)
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(
              values = cols,
              guide = guide_legend(
                          keywidth = 0.5,
                          keyheight = 0.5,
                          label.theme=element_text(size=6)
                 )
            ),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(
              values=cols2,
              guide = guide_legend(
                          keywidth = 0.5,
                          keyheight = 0.5,
                          label.theme=element_text(size=6)
                 )
            ),
        aes_var = response
      ) %>%
      set_scale_theme(
        x = scale_size_continuous(
              range = c(1, 3),
              guide = guide_legend(
                          keywidth = 0.5,
                          keyheight = 0.5,
                          label.theme=element_text(size=6)
                 )
            ),
        aes_var = bray
      )

p2 <- mpseMGS %>%
      mp_plot_dist(
        .distmethod = bray,
        .group = disease,
        group.test = TRUE
      ) +
      scale_color_manual(
        values = c("orange", "#00A08A", "deepskyblue")
      ) +
      scale_fill_manual(
        values = c("orange", "#00A08A", "deepskyblue")
      )

mpseMGS %>%
    mp_adonis(
      .abundance = Abundance,
      .formula = ~ disease + response,
      distmethod = "bray",
      permutation = 9999
    )
```

```
##
## Call:
## vegan::adonis(formula = .formula, data = sampleda, permutations = permutations,      method = distmethod)
##
## Permutation: free
## Number of permutations: 9999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## disease    1    0.4064 0.40638  1.3843 0.03698 0.1563
## response   1    0.3081 0.30807  1.0494 0.02803 0.3946
## Residuals 35   10.2751 0.29357         0.93499
## Total     37   10.9896                 1.00000
```

```
p3 <- mpseMGS %>%
      mp_plot_ord(
        .ord = pcoa,
        .group = disease,
        .size = Observe,
        .starshape = response,
        show.side = FALSE
      ) +
      scale_starshape_manual(values = c(1, 13, 15)) +
      scale_fill_manual(
        values=cols,
        guide=guide_legend(
          keywidth = 0.3,
          keyheight = 0.3,
          label.element = element_text(size = 6),
          override.aes = list(size = 2, starshape = 15)
        )
      ) +
      scale_size_continuous(
        range = c(1, 3),
        guide = guide_legend(
          keywidth = 0.3,
          keyheight = 0.3,
          label.element = element_text(size = 6),
          override.aes = list(starshape = 15)
        )
      )
aplot::plot_list(p1, (aplot::plot_list(p2, p3, nrow=1, widths=c(1, 2))), ncol = 1, heights = c(1.2, 1))
```

2.3.3 Different analysis in MGS level

```
mpseMGS %<>%
    mp_diff_analysis(
       .abundance = Abundance,
       force = TRUE,
       relative = FALSE,
       .group = disease,
       filter.p = "pvalue"
    )
```

```
## The otutree is empty in the MPSE object!
```

```
mpseMGS %>% mp_extract_tree() %>% dplyr::filter(!is.na(pvalue) & pvalue <=0.05, keep.td=FALSE)
```

```
## # A tibble: 15 x 12
##     node label       isTip nodeClass nodeDepth AbundanceBySamp~ LDAupper LDAmean
##    <dbl> <chr>       <lgl> <chr>         <dbl> <list>              <dbl>   <dbl>
##  1    41 s__Alistip~ TRUE  OTU               7 <tibble [38 x 6~     4.15    4.09
##  2    53 s__Clostri~ TRUE  OTU               7 <tibble [38 x 6~     4.25    4.21
##  3    79 s__un_g__O~ TRUE  OTU               7 <tibble [38 x 6~     3.45    3.35
##  4    84 s__Faecali~ TRUE  OTU               7 <tibble [38 x 6~     4.79    4.75
##  5   123 p__Firmicu~ FALSE Phylum            2 <tibble [38 x 6~     4.95    4.92
##  6   132 c__Clostri~ FALSE Class             3 <tibble [38 x 6~     4.93    4.90
##  7   146 o__Clostri~ FALSE Order             4 <tibble [38 x 6~     4.93    4.90
##  8   163 f__Rikenel~ FALSE Family            5 <tibble [38 x 6~     4.15    4.11
##  9   166 f__Clostri~ FALSE Family            5 <tibble [38 x 6~     4.28    4.23
## 10   170 f__Oscillo~ FALSE Family            5 <tibble [38 x 6~     3.45    3.35
## 11   172 f__Ruminoc~ FALSE Family            5 <tibble [38 x 6~     4.76    4.71
## 12   200 g__Alistip~ FALSE Genus             6 <tibble [38 x 6~     4.15    4.11
## 13   204 g__Clostri~ FALSE Genus             6 <tibble [38 x 6~     4.28    4.23
## 14   214 g__Oscilli~ FALSE Genus             6 <tibble [38 x 6~     3.45    3.35
## 15   217 g__Faecali~ FALSE Genus             6 <tibble [38 x 6~     4.79    4.75
## # ... with 4 more variables: LDAlower <dbl>, Sign_disease <chr>, pvalue <dbl>,
## #   fdr <dbl>
```

```r
library(forcats)

trda <- mpseMGS %>% mp_extract_tree()

p <- ggtree(trda, layout = 'radial') +
     geom_tiplab(size = 1.8, offset = 11) +
     geom_hilight(
         mapping = aes(
           subset = nodeClass == "Phylum",
           node = node,
           fill = label
         )
     )
p2 <- p +
      ggnewscale::new_scale_fill() +
      geom_fruit(
         data = td_unnest(AbundanceBySample, names_repair=tidyr::tidyr_legacy),
         geom = geom_star,
         mapping = aes(
            x = fct_reorder(Sample, disease, .fun=min),
            size = Abundance,
            fill = disease,
            subset = Abundance > 0
         ),
         starshape = 13,
         offset = 0.02,
         pwidth = 1,
         grid.params = list(linetype=2)
      ) +
      scale_size_continuous(name="Relative Abundance (%)",range = c(1, 3)) +
      scale_fill_manual(values = cols)

p3 <- p2 +
      ggnewscale::new_scale("fill") +
      geom_fruit(
         geom = geom_col,
         mapping = aes(
                       x = LDAmean,
                       fill = Sign_disease,
                       subset = !is.na(LDAmean)
                       ),
         orientation = "y",
         offset = .05,
         pwidth = 0.5,
         width = 0.5, # the parameter of geom_col
         axis.params = list(axis = "x",
                            title = "Log10(LDA)",
                            title.height = 0.001,
                            title.size = 2,
                            text.size = 1.8,
                            vjust = 1),
         grid.params = list(linetype = 1)
      ) +
      ggnewscale::new_scale("size") +
      geom_point(
         data=td_filter(!is.na(pvalue)),
         mapping = aes(size = -log10(pvalue),
                       fill = Sign_disease
                   ),
         shape = 21
      ) +
      scale_size_continuous(range=c(1, 3)) +
      scale_fill_manual(values=cols) +
      theme(
           legend.key.height = unit(0.3, "cm"),
           legend.key.width = unit(0.3, "cm"),
           legend.spacing.y = unit(0.02, "cm"),
           legend.text = element_text(size = 7),
           legend.title = element_text(size = 9),
      )
p3
```

3 The analysis of the mosquito ecology data using MicrobiotaProcess

MicrobiotaProcess also can be used to perform the other related ecology data analysis, besides the microbial community data. Here, we used an example data about a Mosquito ecology study (REISKIND et al. 2017) to show how to use MicrobiotaProcess to perform the analysis of the related ecology study. The data was obtained from the github[3].

3.1 Loading data and Construction of MPSE object

The 1 to 14 columns are the sample metadata including the study site, and habitat, etc. and the others columns represent the abundance of mosquito species the in each sample. In details, you can refer to the blog[4]

```
data <- read.csv("./data/Mosquito_ecology/data.csv", row.names=1)
abun.d <- data[, 14:36]
sample.d <- data[, 1:13]
# We implements `MPSE` function to build the `MPSE` object, which requires the abundance table (matrix-like).
mpse <- MPSE(assays=list(Abundance=t(abun.d)), colData=sample.d)
mpse
```

```
## # A MPSE-tibble (MPSE object) abstraction: 1,035 x 16
## # OTU=23 | Samples=45 | Assays=Abundance | Taxonomy=NULL
##    OTU      Sample Abundance Region Transect Habitat DeciduousForest
##    <chr>    <chr>      <int> <chr>  <chr>    <chr>             <dbl>
##  1 Cx.sal   DU1.1         19 Durham DU1      Field              125.
##  2 Ae.albo  DU1.1          0 Durham DU1      Field              125.
##  3 Ae.cin   DU1.1          1 Durham DU1      Field              125.
##  4 Ae.vex   DU1.1         16 Durham DU1      Field              125.
```

[3] https://github.com/rgriff23/Mosquito_ecology

[4] http://www.randigriffin.com/2017/05/23/mosquito-community-ecology-in-vegan.html

```
##  5 Ps.fer   DU1.1          1 Durham DU1      Field                125.
##  6 Cx.err   DU1.1        372 Durham DU1      Field                125.
##  7 Ps.col   DU1.1        104 Durham DU1      Field                125.
##  8 Ae.tris  DU1.1          0 Durham DU1      Field                125.
##  9 Cx.pip.q DU1.1          2 Durham DU1      Field                125.
## 10 Ae.can   DU1.1          0 Durham DU1      Field                125.
## # ... with 1,025 more rows, and 9 more variables: EvergreenForest <dbl>,
## #   Grassland <dbl>, MixedForest <dbl>, ShrubScrub <dbl>, BarrenLand <dbl>,
## #   Building <dbl>, Pavement <dbl>, CultivatedCrops <dbl>, TrapNights <int>
```

3.2 Alpha diversity analysis of the Mosquito ecology study

The MicrobiotaProcess provides some verbs of dplyr, which allows user to explore the MPSE class effectively and develop reproducible and human-readable pipelines

```r
cols = c("floralwhite", "lightgoldenrod1","chartreuse2", "chartreuse4", "darkgreen")
# Adjusting the order of Habitat
mpse %<>%
   dplyr::mutate(
     Habitat = factor(
       Habitat,
       levels = c("Field", "NearField", "Edge", "NearForest", "Forest")
    )
   )
mpse
```

```
## # A MPSE-tibble (MPSE object) abstraction: 1,035 x 16
## # OTU=23 | Samples=45 | Assays=Abundance | Taxanomy=NULL
##    OTU      Sample Abundance Region Transect Habitat DeciduousForest
##    <chr>    <chr>      <int> <chr>  <chr>    <fct>             <dbl>
##  1 Cx.sal   DU1.1         19 Durham DU1      Field              125.
##  2 Ae.albo  DU1.1          0 Durham DU1      Field              125.
##  3 Ae.cin   DU1.1          1 Durham DU1      Field              125.
##  4 Ae.vex   DU1.1         16 Durham DU1      Field              125.
##  5 Ps.fer   DU1.1          1 Durham DU1      Field              125.
##  6 Cx.err   DU1.1        372 Durham DU1      Field              125.
##  7 Ps.col   DU1.1        104 Durham DU1      Field              125.
##  8 Ae.tris  DU1.1          0 Durham DU1      Field              125.
##  9 Cx.pip.q DU1.1          2 Durham DU1      Field              125.
## 10 Ae.can   DU1.1          0 Durham DU1      Field              125.
## # ... with 1,025 more rows, and 9 more variables: EvergreenForest <dbl>,
## #   Grassland <dbl>, MixedForest <dbl>, ShrubScrub <dbl>, BarrenLand <dbl>,
## #   Building <dbl>, Pavement <dbl>, CultivatedCrops <dbl>, TrapNights <int>
```

```r
# force=TRUE meaning the Abundance will be used to calculate the alpha index without rarefaction
mpse %<>% mp_cal_alpha(.abundance=Abundance, force=TRUE)
# test the relationship between the Observe Species and Habitat or Shannon and Habitat.
mpse %>% mp_extract_sample() %>% lm(formula=Observe ~ Habitat, data=.) %>% anova()
```

```
## Analysis of Variance Table
##
## Response: Observe
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Habitat    4   57.2   14.30  2.9485 0.03164 *
## Residuals 40  194.0    4.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mpse %>% mp_extract_sample() %>% lm(formula=Shannon ~ Habitat, data=.) %>% anova()
```

```
## Analysis of Variance Table
##
```

```
## Response: Shannon
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Habitat    4 1.7619 0.44048   2.395 0.06639 .
## Residuals 40 7.3565 0.18391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result of ANOVA test revealed that the richness of the mosquito species was significantly associated with the **habitat**. Then the result was visualized by *mp_plot_alpha*.

```
p.alpha <- mpse %>%
     mp_plot_alpha(.group = Habitat, .alpha = c(Observe, Shannon), test = NULL) +
     scale_fill_manual(values = cols) +
     scale_color_manual(values = cols) +
     theme(legend.position = "none")
p.alpha
```

3.3 Beta Diversity Analysis of the Mosquito ecology study

Here, we use the cca (constrained correspondence analysis) to test which environment factor is related to the Mosquito species in the habitat.

```
mpse %<>%
    mutate(NormAbun=sqrt(Abundance)/TrapNights) %>%
    mp_cal_cca(
       .abundance  = NormAbun,
       .formula = ~(DeciduousForest+
           EvergreenForest+
           Grassland+
           MixedForest+
           ShrubScrub)+
           Condition(
             BarrenLand+
             Building+
             Pavement+
             CultivatedCrops
           )
    )
mpse
```

```
## # A MPSE-tibble (MPSE object) abstraction: 1,035 x 26
## # OTU=23 | Samples=45 | Assays=Abundance, NormAbun | Taxanomy=NULL
##    OTU      Sample Abundance NormAbun Region Transect Habitat DeciduousForest
##    <chr>    <chr>      <int>    <dbl> <chr>  <chr>    <fct>             <dbl>
##  1 Cx.sal   DU1.1         19    0.436 Durham DU1      Field              125.
##  2 Ae.albo  DU1.1          0    0     Durham DU1      Field              125.
##  3 Ae.cin   DU1.1          1    0.1   Durham DU1      Field              125.
##  4 Ae.vex   DU1.1         16    0.4   Durham DU1      Field              125.
##  5 Ps.fer   DU1.1          1    0.1   Durham DU1      Field              125.
##  6 Cx.err   DU1.1        372    1.93  Durham DU1      Field              125.
##  7 Ps.col   DU1.1        104    1.02  Durham DU1      Field              125.
##  8 Ae.tris  DU1.1          0    0     Durham DU1      Field              125.
##  9 Cx.pip.q DU1.1          2    0.141 Durham DU1      Field              125.
## 10 Ae.can   DU1.1          0    0     Durham DU1      Field              125.
## # ... with 1,025 more rows, and 18 more variables: EvergreenForest <dbl>,
## #   Grassland <dbl>, MixedForest <dbl>, ShrubScrub <dbl>, BarrenLand <dbl>,
## #   Building <dbl>, Pavement <dbl>, CultivatedCrops <dbl>, TrapNights <int>,
## #   Observe <dbl>, Chao1 <dbl>, ACE <dbl>, Shannon <dbl>, Simpson <dbl>,
## #   Pielou <dbl>, CCA1 (25.28%) <dbl>, CCA2 (7.34%) <dbl>, CCA3 (3.39%) <dbl>
```

The raw result of pCCA was added the *internal_attr*, which can be extract by *mp_extract_internal_attr* with specific *name=cca*. Then it can be performed the significance test using the functions of *vegan* (Oksanen et al. 2020), such as *anova.cca*, *permutest*.

```
# Extract the raw result of cca analysis
# And significance test with anova
mpse %>%
    mp_extract_internal_attr(name=cca) %>%
    anova()
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = x ~ (DeciduousForest + EvergreenForest + Grassland + MixedForest + ShrubScrub) + Condit
##          Df ChiSquare      F Pr(>F)
## Model     5   0.38999 4.4365  0.001 ***
## Residual 35   0.61534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Further we used *mp_envfit* to identity the environment variables that were significantly associated with the mosquito communities.

```
# fits environmental vectors onto cca
mpse %<>%
    mp_envfit(
       .ord = cca,
       .env = c(
          DeciduousForest,
          EvergreenForest,
          Grassland,
          MixedForest,
          ShrubScrub
        ),
       action = "add",
       permutation = 9999
    )

# Extract the raw result of envfit analysis
mpse %>% mp_extract_internal_attr(name=cca_envfit)
```

```
##
## ***VECTORS
##
##                      CCA1     CCA2     CCA3     r2 Pr(>r)
## DeciduousForest  0.42979  0.90272 -0.01945 0.3804 0.0020 **
## EvergreenForest  0.91539 -0.34612 -0.20559 0.5557 0.0001 ***
## Grassland       -0.97679 -0.21356  0.01639 0.7216 0.0001 ***
## MixedForest      0.77120 -0.25826  0.58186 0.1936 0.0929 .
## ShrubScrub      -0.73942  0.22976 -0.63283 0.2595 0.0537 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 9999
```

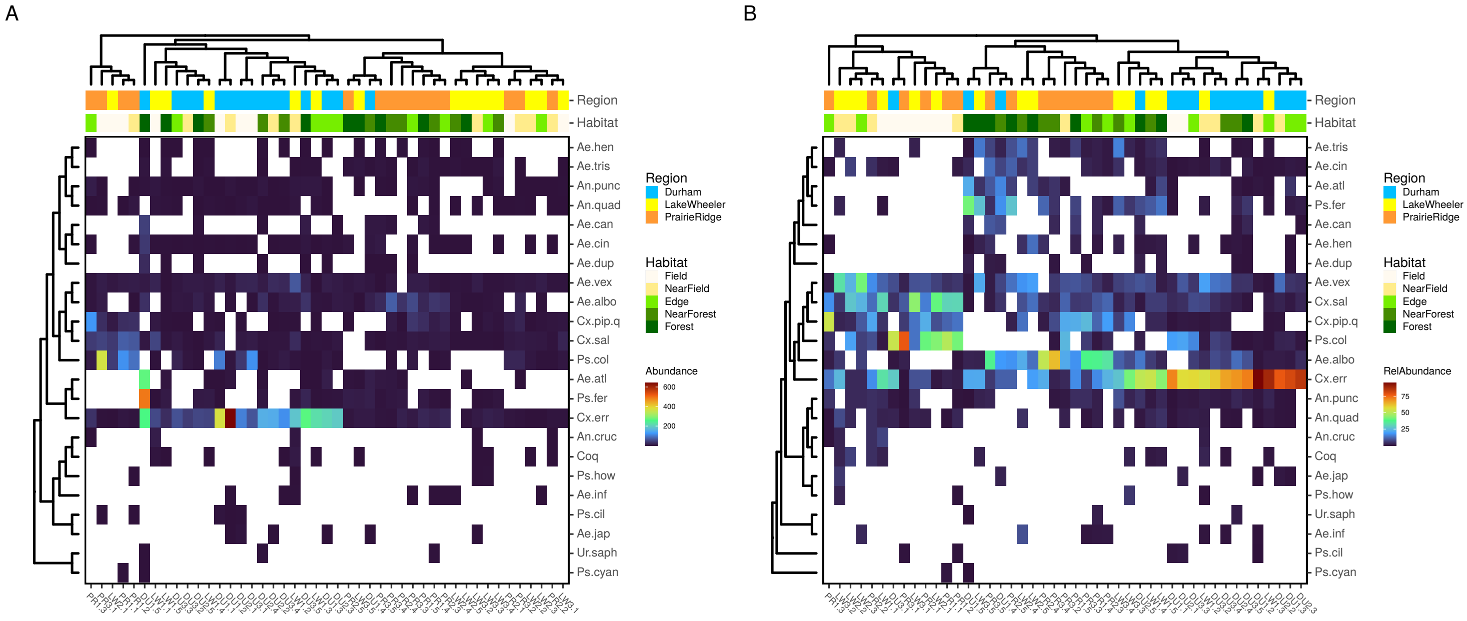
Then we used *mp_plot_ord* to visualize the result of pCCA.

```
# visualization only pCCA
f <- mpse %>%
     mp_plot_ord(
       .ord = cca,
       .group = Habitat,
       .size = Observe,
       .starshape = Region,
       show.side = FALSE,
       show.envfit = FALSE,
       colour = 'white',
       bg.colour = 'black'
     ) +
     scale_starshape_manual(values=c(1, 13, 15)) +
     scale_fill_manual(
        values = cols,
        guide = guide_legend(
          override.aes = list(starshape=15)
        )
     ) +
     scale_size_continuous(
       range = c(1, 3),
       guide = guide_legend(override.aes = list(starshape=15))
     ) +
     theme(
        legend.key.height = unit(0.3, "cm"),
        legend.key.width = unit(0.3, "cm"),
        legend.spacing.y = unit(0.02, "cm"),
        legend.text = element_text(size = 7),
        legend.title = element_text(size = 9),
     )

# visualization with envfit result
p <- mpse %>%
     mp_plot_ord(
       .ord = cca,
       .group = Habitat,
       .size = Observe,
       .starshape = Region,
       show.side = FALSE,
       show.envfit = TRUE,
       colour = "white",
       bg.colour = "black"
     ) +
     scale_starshape_manual(values=c(1, 13, 15)) +
     scale_fill_manual(
        values = cols,
        guide = guide_legend(
          override.aes = list(starshape=15)
        )
     ) +
     scale_size_continuous(
       range = c(1, 3),
       guide = guide_legend(override.aes = list(starshape=15))
     ) +
     theme(
        legend.key.height = unit(0.3, "cm"),
        legend.key.width = unit(0.3, "cm"),
        legend.spacing.y = unit(0.02, "cm"),
        legend.text = element_text(size = 7),
        legend.title = element_text(size = 9),
     )

aplot::plot_list(f, p, tag_levels="A")
```

3.4 The distribution of Mosquito species in the study.

We used *mp_cal_abundance* and *mp_plot_abundance* to calculated and visualized the abundance of the Mosquito species in the study.

```
cols2 <- c("deepskyblue", "yellow", "#FF9933")

# The theme and scale of fill of heatmap
Abund.char <- list(
           scale_fill_viridis_c(option = "H"),
           theme(
             axis.text.x = element_text(size = 6),
             axis.text.y = element_text(size = 8),
             legend.title = element_text(size = 7),
             legend.text = element_text(size = 5),
             legend.key.width = unit(0.3, "cm"),
             legend.key.height = unit(0.3, "cm")
           )
      )

# The theme and legend of annotate bar of 'Habitat' variable
Habitat.char <- list(
           scale_fill_manual(values = cols),
           theme(
             legend.key.height = unit(0.3, "cm"),
             legend.key.width = unit(0.3, "cm"),
             legend.spacing.y = unit(0.02, "cm"),
             legend.text = element_text(size = 7),
             legend.title = element_text(size = 9)
           )
      )

# The theme and legend of annotate bar of 'Region' variable
Region.char <- list(
           scale_fill_manual(values = cols2),
           theme(
             legend.key.height = unit(0.3, "cm"),
             legend.key.width = unit(0.3, "cm"),
             legend.spacing.y = unit(0.02, "cm"),
             legend.text = element_text(size = 7),
             legend.title = element_text(size = 9)
           )
      )

# visualization of the count abundance.
p.count <- mpse %>%
    mp_cal_abundance(
      .abundance = Abundance,
      force = T,
      relative = F
    ) %>%
    mp_plot_abundance(
      .abundance = Abundance,
      force = T,
      relative = F,
      geom = "heatmap",
      topn = "all",
      .group = c(Habitat, Region)
    ) %>%
    set_scale_theme(
      x = Abund.char,
      aes_var = Abundance
    ) %>%
    set_scale_theme(
      x = Habitat.char,
      aes_var = Habitat
    ) %>%
    set_scale_theme(
      x = Region.char,
      aes_var = Region
    )

# visualization of the relative abundance
p.rel <- mpse %>%
    mp_cal_abundance(
      .abundance = Abundance,
      force = T,
      relative = T
    ) %>%
    mp_plot_abundance(
      .abundance = Abundance,
      force = T,
      relative = T,
      geom = "heatmap",
      topn = "all",
      .group = c(Habitat, Region)
    ) %>%
    set_scale_theme(
      x = Abund.char,
      aes_var = RelAbundance
    ) %>%
    set_scale_theme(
      x = Habitat.char,
      aes_var = Habitat
    ) %>%
    set_scale_theme(
      x = Region.char,
      aes_var = Region
    )

aplot::plot_list(p.count, p.rel, tag_levels="A")
```



Then We can use *mp_diff_analysis* to identify the species that have significant difference abundance between the **field** and **forest**. We found the Cx.sal (*Culex salinarius*) and Ps.col (*Psorophora columbiae*) were significantly enriched in **field**, However, the Ae.albo (*Aedes albopicta*), Ae.cin (*Aedes cinereus*), Ps.fer (*Psorophora ferox*), Ae.tris (*Aedes triseriatus*), Ae.can (*Aedes canadensis*), Ae.hen (*Aedes hendersoni*), Ae.atl (*Aedes atlanticus*) and Ae.dup (*Aedes dupreei*) were significantly enriched in the **forest**

```
mpse %>%
    dplyr::filter(Habitat %in% c("Field", "Forest")) %>%
    dplyr::mutate(Habitat = as.vector(Habitat)) %>%
    mp_diff_analysis(.abundance=Abundance, force=T, relative=T, .group=Habitat) %>%
    mp_extract_feature() %>%
    dplyr::filter(fdr<=0.05) %>%
    print(width=200)
```

```
## The otutree is empty in the MPSE object!
```

```
## The taxatree is empty in the MPSE object!
```

```
## # A tibble: 10 x 8
##    OTU     AbundanceBySample LDAupper LDAmean LDAlower Sign_Habitat   pvalue
##    <chr>   <list>               <dbl>   <dbl>    <dbl> <chr>           <dbl>
##  1 Cx.sal  <tibble [18 x 6]>     4.96    4.92     4.87 Field        0.00705
##  2 Ae.albo <tibble [18 x 6]>     4.83    4.79     4.75 Forest       0.000229
##  3 Ae.cin  <tibble [18 x 6]>     4.36    4.31     4.25 Forest       0.0159
##  4 Ps.fer  <tibble [18 x 6]>     4.94    4.90     4.87 Forest       0.00122
##  5 Ps.col  <tibble [18 x 6]>     5.26    5.24     5.22 Field        0.000327
##  6 Ae.tris <tibble [18 x 6]>     4.49    4.46     4.43 Forest       0.000530
##  7 Ae.can  <tibble [18 x 6]>     4.28    4.24     4.19 Forest       0.0119
##  8 Ae.hen  <tibble [18 x 6]>     4.28    4.23     4.18 Forest       0.000483
##  9 Ae.atl  <tibble [18 x 6]>     4.59    4.56     4.52 Forest       0.00311
## 10 Ae.dup  <tibble [18 x 6]>     4.03    3.96     3.88 Forest       0.0119
##        fdr
##      <dbl>
##  1 0.0211
##  2 0.00278
##  3 0.0334
##  4 0.00513
##  5 0.00278
##  6 0.00278
##  7 0.0278
##  8 0.00278
##  9 0.0109
## 10 0.0278
```

4 Session information

Here is the output of sessionInfo() on the system on which this document was compiled:

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.1 (2021-08-10)
##  os       Ubuntu 18.04.4 LTS
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Asia/Shanghai
##  date     2021-12-15
##
## - Packages -------------------------------------------------------------------
##  package              * version   date       lib
##  ade4                   1.7-18    2021-09-16 [1]
##  ape                    5.5-3     2021-12-03 [1]
##  aplot                * 0.1.1     2021-09-22 [1]
##  assertthat             0.2.1     2019-03-21 [1]
##  Biobase              * 2.54.0    2021-10-26 [1]
##  BiocGenerics         * 0.40.0    2021-10-26 [1]
##  BiocManager            1.30.16   2021-06-15 [1]
##  biomformat             1.22.0    2021-10-26 [1]
##  Biostrings             2.62.0    2021-10-26 [1]
##  bitops                 1.0-7     2021-04-24 [1]
##  bookdown               0.24      2021-09-02 [1]
##  cachem                 1.0.6     2021-08-19 [1]
##  class                  7.3-19    2021-05-03 [1]
##  classInt               0.4-3     2020-04-07 [1]
##  cli                    3.1.0     2021-10-27 [1]
##  cluster                2.1.2     2021-04-17 [1]
##  codetools              0.2-18    2020-11-04 [1]
##  coin                 * 1.4-2     2021-10-08 [1]
##  colorspace             2.0-2     2021-06-24 [1]
##  conflicted           * 1.0.4     2019-06-21 [1]
##  corrr                  0.4.3     2020-11-24 [1]
##  crayon                 1.4.2     2021-10-29 [1]
##  data.table             1.14.2    2021-09-27 [1]
##  DBI                    1.1.1     2021-01-15 [1]
##  DelayedArray           0.20.0    2021-10-26 [1]
##  digest                 0.6.28    2021-09-23 [1]
##  dplyr                  1.0.7     2021-06-18 [1]
##  dtplyr                 1.1.0     2021-02-20 [1]
##  e1071                  1.7-9     2021-09-16 [1]
##  edgeR                * 3.36.0    2021-10-26 [1]
##  ellipsis               0.3.2     2021-04-29 [1]
##  evaluate               0.14      2019-05-28 [1]
##  fansi                  0.5.0     2021-05-25 [1]
##  farver                 2.1.0     2021-02-28 [1]
##  fastmap                1.1.0     2021-01-25 [1]
##  forcats              * 0.5.1     2021-01-27 [1]
##  foreach                1.5.1     2020-10-15 [1]
##  generics               0.1.1     2021-10-25 [1]
##  GenomeInfoDb         * 1.30.0    2021-10-26 [1]
##  GenomeInfoDbData       1.2.7     2021-10-29 [1]
##  GenomicRanges        * 1.46.0    2021-10-26 [1]
##  ggalluvial             0.12.3    2020-12-05 [1]
##  ggfun                  0.0.4     2021-09-17 [1]
##  ggh4x                  0.2.0     2021-08-21 [1]
##  gghalves               0.1.1     2020-11-08 [1]
##  ggnewscale           * 0.4.5     2021-01-11 [1]
##  ggplot2              * 3.3.5     2021-06-25 [1]
##  ggplotify              0.1.0     2021-09-02 [1]
##  ggpp                 * 0.4.2     2021-07-31 [1]
##  ggrepel              * 0.9.1     2021-01-15 [1]
##  ggside                 0.1.2     2021-07-21 [1]
##  ggsignif               0.6.3     2021-09-09 [1]
##  ggstar               * 1.0.3     2021-12-03 [1]
##  ggtree               * 3.3.0.901 2021-12-10 [1]
##  ggtreeExtra          * 1.5.1     2021-11-24 [1]
##  ggupset                0.3.0     2020-05-05 [1]
##  ggVennDiagram          1.1.4     2021-07-07 [1]
##  glue                   1.5.0     2021-11-07 [1]
##  gridExtra              2.3       2017-09-09 [1]
##  gridGraphics            0.5-1     2020-12-13 [1]
##  gtable                  0.3.0     2019-03-25 [1]
##  hms                     1.1.1     2021-09-26 [1]
##  htmltools               0.5.2     2021-08-25 [1]
##  httr                    1.4.2     2020-07-20 [1]
##  igraph                  1.2.7     2021-10-15 [1]
##  IRanges               * 2.28.0    2021-10-26 [1]
##  iterators               1.0.13    2020-10-15 [1]
##  jsonlite                1.7.2     2020-12-09 [1]
##  kableExtra            * 1.3.4     2021-02-20 [1]
##  KernSmooth              2.23-20   2021-05-03 [1]
##  knitr                   1.36      2021-09-29 [1]
##  labeling                0.4.2     2020-10-20 [1]
##  lattice                 0.20-45   2021-09-22 [1]
##  lazyeval                0.2.2     2019-03-15 [1]
##  libcoin                 1.0-9     2021-09-27 [1]
##  lifecycle               1.0.1     2021-09-24 [1]
##  limma                 * 3.50.0    2021-10-26 [1]
##  locfit                  1.5-9.4   2020-03-25 [1]
##  magrittr              * 2.0.1     2020-11-17 [1]
##  MASS                    7.3-54    2021-05-03 [1]
##  Matrix                  1.3-4     2021-06-01 [1]
##  MatrixGenerics        * 1.6.0     2021-10-26 [1]
##  matrixStats           * 0.61.0    2021-09-17 [1]
##  mgcv                    1.8-38    2021-10-06 [1]
##  MicrobiotaProcess     * 1.7.4     2021-12-15 [1]
##  modeltools              0.2-23    2020-03-05 [1]
##  multcomp                1.4-17    2021-04-29 [1]
##  multtest                2.50.0    2021-10-26 [1]
##  munsell                 0.5.0     2018-06-12 [1]
##  mvtnorm                 1.1-3     2021-10-08 [1]
##  nlme                    3.1-153   2021-09-07 [1]
##  patchwork             * 1.1.1     2020-12-17 [1]
##  permute                 0.9-5     2019-03-12 [1]
##  phyloseq                1.38.0    2021-10-26 [1]
##  pillar                  1.6.4     2021-10-18 [1]
##  pkgconfig               2.0.3     2019-09-22 [1]
##  plyr                    1.8.6     2020-03-03 [1]
##  preprocessCore          1.56.0    2021-10-26 [1]
##  proxy                   0.4-26    2021-06-07 [1]
##  purrr                   0.3.4     2020-04-17 [1]
##  R6                      2.5.1     2021-08-19 [1]
##  RColorBrewer            1.1-2     2014-12-07 [1]
##  Rcpp                    1.0.7     2021-07-07 [1]
##  RCurl                   1.98-1.5  2021-09-17 [1]
##  readr                   2.0.2     2021-09-27 [1]
##  reshape2                1.4.4     2020-04-09 [1]
##  rhdf5                   2.38.0    2021-10-26 [1]
##  rhdf5filters            1.6.0     2021-10-26 [1]
##  Rhdf5lib                1.16.0    2021-10-26 [1]
##  rlang                   0.4.12    2021-10-18 [1]
##  rmarkdown               2.11      2021-09-14 [1]
##  rstudioapi              0.13      2020-11-12 [1]
##  rvcheck               * 0.2.0     2021-09-14 [1]
##  RVenn                   1.1.0     2019-07-18 [1]
##  rvest                   1.0.2     2021-10-16 [1]
##  S4Vectors             * 0.32.0    2021-10-26 [1]
##  sandwich                3.0-1     2021-05-18 [1]
##  scales                  1.1.1     2020-05-11 [1]
##  sessioninfo             1.1.1     2018-11-05 [1]
##  sf                     1.0-3     2021-10-07 [1]
##  shadowtext           * 0.0.9     2021-09-19 [1]
##  stringi                1.7.5     2021-10-04 [1]
##  stringr                1.4.0     2019-02-10 [1]
##  SummarizedExperiment * 1.24.0    2021-10-26 [1]
##  survival             * 3.2-13    2021-08-24 [1]
##  svglite                2.0.0     2021-02-20 [1]
##  systemfonts            1.0.3     2021-10-13 [1]
##  TH.data                1.1-0     2021-09-27 [1]
##  tibble                 3.1.6     2021-11-07 [1]
##  tidybulk             * 1.6.1     2021-10-31 [1]
##  tidyr                  1.1.4     2021-09-27 [1]
##  tidyselect             1.1.1     2021-04-30 [1]
##  tidytree               0.3.6     2021-11-12 [1]
##  treeio                 1.18.0    2021-10-26 [1]
##  tzdb                   0.1.2     2021-07-20 [1]
##  units                  0.7-2     2021-06-08 [1]
##  utf8                   1.2.2     2021-07-24 [1]
##  vctrs                  0.3.8     2021-04-29 [1]
##  vegan                  2.5-7     2020-11-28 [1]
##  viridisLite            0.4.0     2021-04-13 [1]
##  webshot                0.5.2     2019-11-22 [1]
##  wget                 * 0.0.1     2021-12-06 [1]
##  withr                  2.4.2     2021-04-18 [1]
##  xfun                   0.28      2021-11-04 [1]
##  xml2                   1.3.2     2020-04-23 [1]
##  XVector                0.34.0    2021-10-26 [1]
##  yaml                   2.2.1     2020-02-01 [1]
##  yulab.utils            0.0.4.901 2021-11-17 [1]
##  zlibbioc               1.40.0    2021-10-26 [1]
##  zoo                    1.8-9     2021-03-09 [1]
##  source
##  CRAN (R 4.1.1)
##  Github (emmanuelparadis/ape@decfeac)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.1)
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Github (xiangpin/ggstar@5e1f018)
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  local
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.1)
##  Github (YuLab-SMU/yulab.utils@98ec2db)
##  Bioconductor
##  CRAN (R 4.1.1)
##
## [1] /mnt/d/UbuntuApps/R/4.1.1/lib/R/library
```

References

Huang, Ruizhu, Charlotte Soneson, Felix G. M. Ernst, Kevin C. Rue-Albrecht, Guangchuang Yu, Stephanie C. Hicks, and Mark D. Robinson. 2021. “TreeSummarizedExperiment: A S4 Class for Data with Hierarchical Structure.” *F1000Research* 9: 1246. https://f1000research.com/articles/9-1246.

Mangiola, Stefano, Ramyar Molania, Ruining Dong, Maria A. Doyle, and Anthony T. Papenfuss. 2021. “Tidybulk: An R Tidy Framework for Modular Transcriptomic Data Analysis.” *Genome Biology* 22 (1): 42. https://doi.org/10.1186/s13059-020-02233-7.

McMurdie, Paul J., and Joseph N Paulson. 2021. *Biomformat: An Interface Package for the Biom File Format.*

McMurdie, Susan, Paul J. AND Holmes. 2013. “Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data.” *PLOS ONE* 8 (4): 1–11. https://doi.org/10.1371/journal.pone.0061217.

Morgan, Martin, Valerie Obenchain, Jim Hester, and Hervé Pagès. 2021. *SummarizedExperiment: SummarizedExperiment Container*. https://bioconductor.org/packages/SummarizedExperiment.

Oksanen, Jari, F. Guillaume Blanchet, Michael Friendly, Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R. Minchin, et al. 2020. *Vegan: Community Ecology Package*. https://CRAN.R-project.org/package=vegan.

REISKIND, M. H., R. H. GRIFFIN, M. S. JANAIRO, and K. A. HOPPERSTAD. 2017. “Mosquitoes of Field and Forest: The Scale of Habitat Segregation in a Diverse Mosquito Assemblage.” *Medical and Veterinary Entomology* 31 (1): 44–54. https://doi.org/https://doi.org/10.1111/mve.12193.

Research Network Consortium, Integrative HMP (iHMP). 2014. “The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles During Periods of Human Health and Disease.” *Cell Host & Microbe* 16 (3): 276–89. https://doi.org/10.1016/j.chom.2014.08.014.

Segata, Nicola, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. 2012. “Metagenomic Microbial Community Profiling Using Unique Clade-Specific Marker Genes.” *Nature Methods* 9 (8): 811–14. https://doi.org/10.1038/nmeth.2066.

Wickham, Hadley. 2011. “Ggplot2.” *WIREs Computational Statistics* 3 (2): 180–85. https://doi.org/https://doi.org/10.1002/wics.147.

Xu, Shuangbin, Zehan Dai, Pingfan Guo, Xiaocong Fu, Shanshan Liu, Lang Zhou, Wenli Tang, et al. 2021. “ggtreeExtra: Compact Visualization of Richly Annotated Phylogenetic Data.” *Molecular Biology and Evolution* 38 (9): 4039–42. https://doi.org/10.1093/molbev/msab166.

Yu, Guangchuang, David K. Smith, Huachen Zhu, Yi Guan, and Tommy Tsan-Yuk Lam. 2017. “Ggtree: An R Package for Visualization and Annotation of Phylogenetic Trees with Their Covariates and Other Associated Data.” *Methods in Ecology and Evolution* 8 (1): 28–36. https://doi.org/https://doi.org/10.1111/2041-210X.12628.

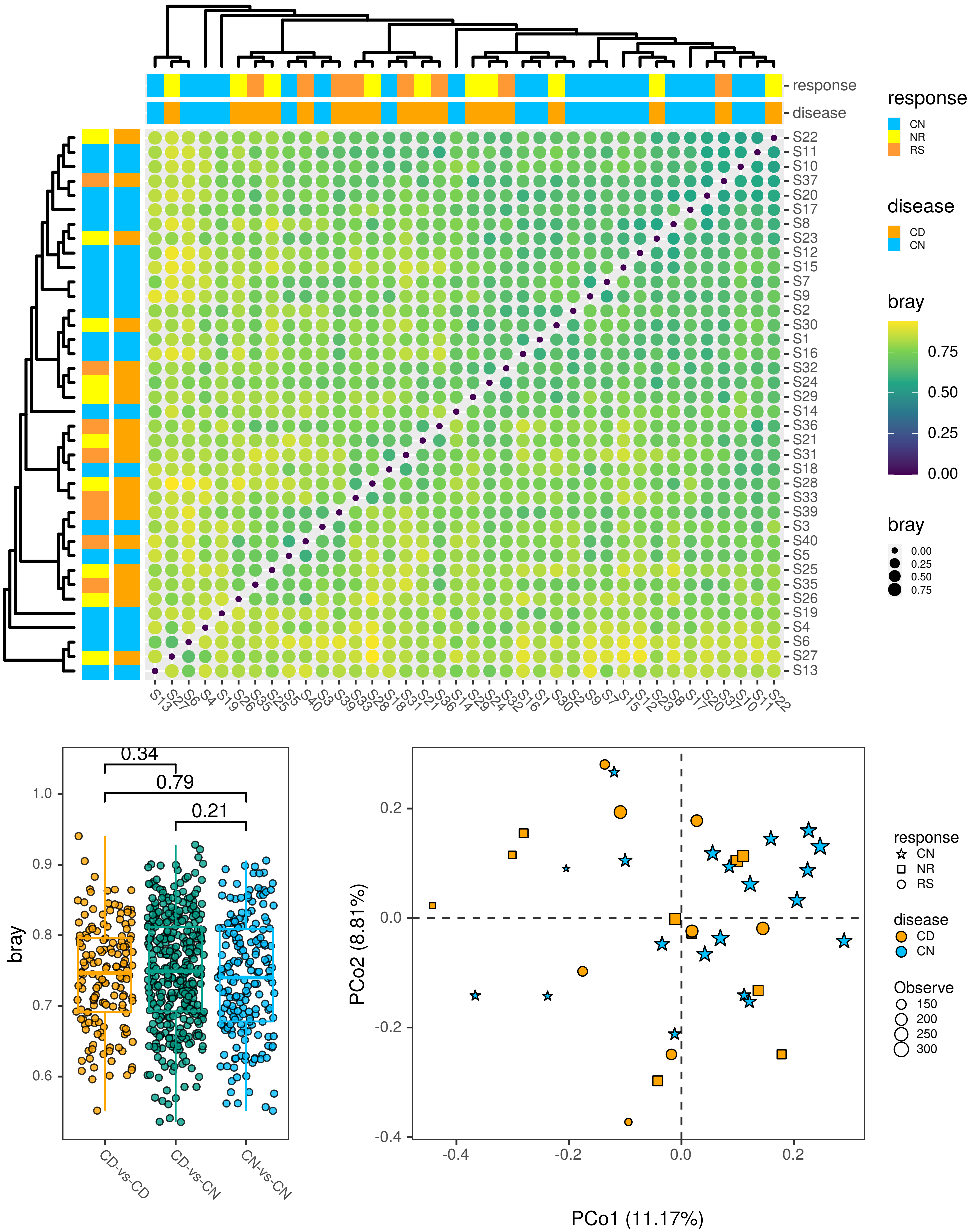


Fig. S15: **The distance heatmap and boxplot and the PCoA plot**

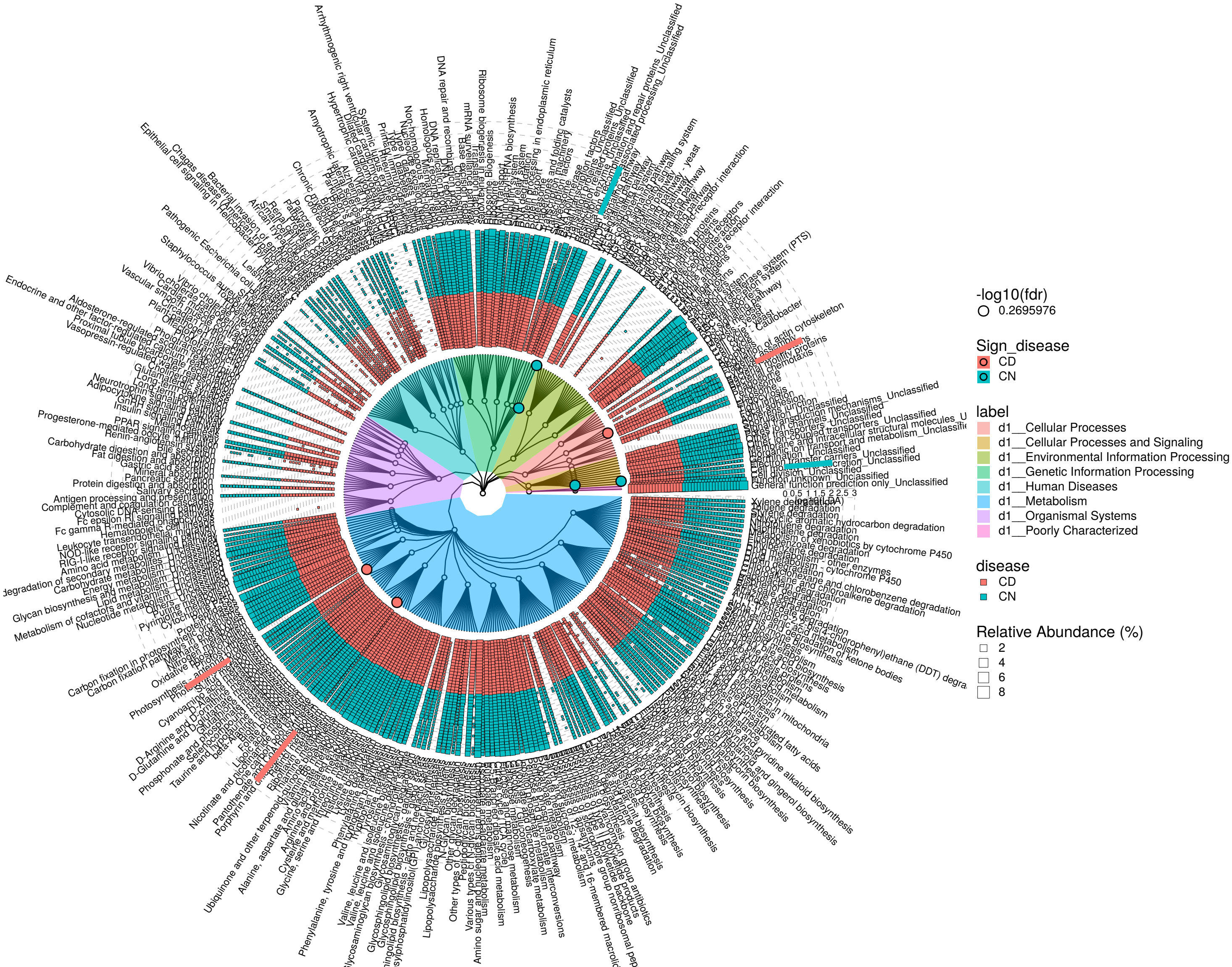


Fig. S16: **The result of the different analysis based on the KEGG pathway data**

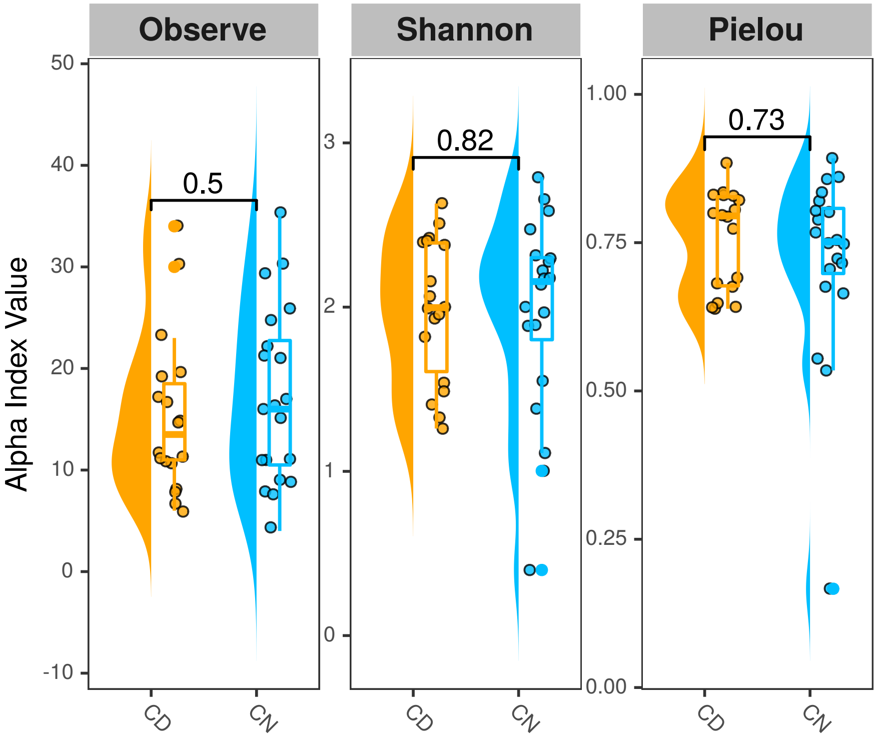


Fig. S17: **The alpha diversity boxplot based on MGS data**

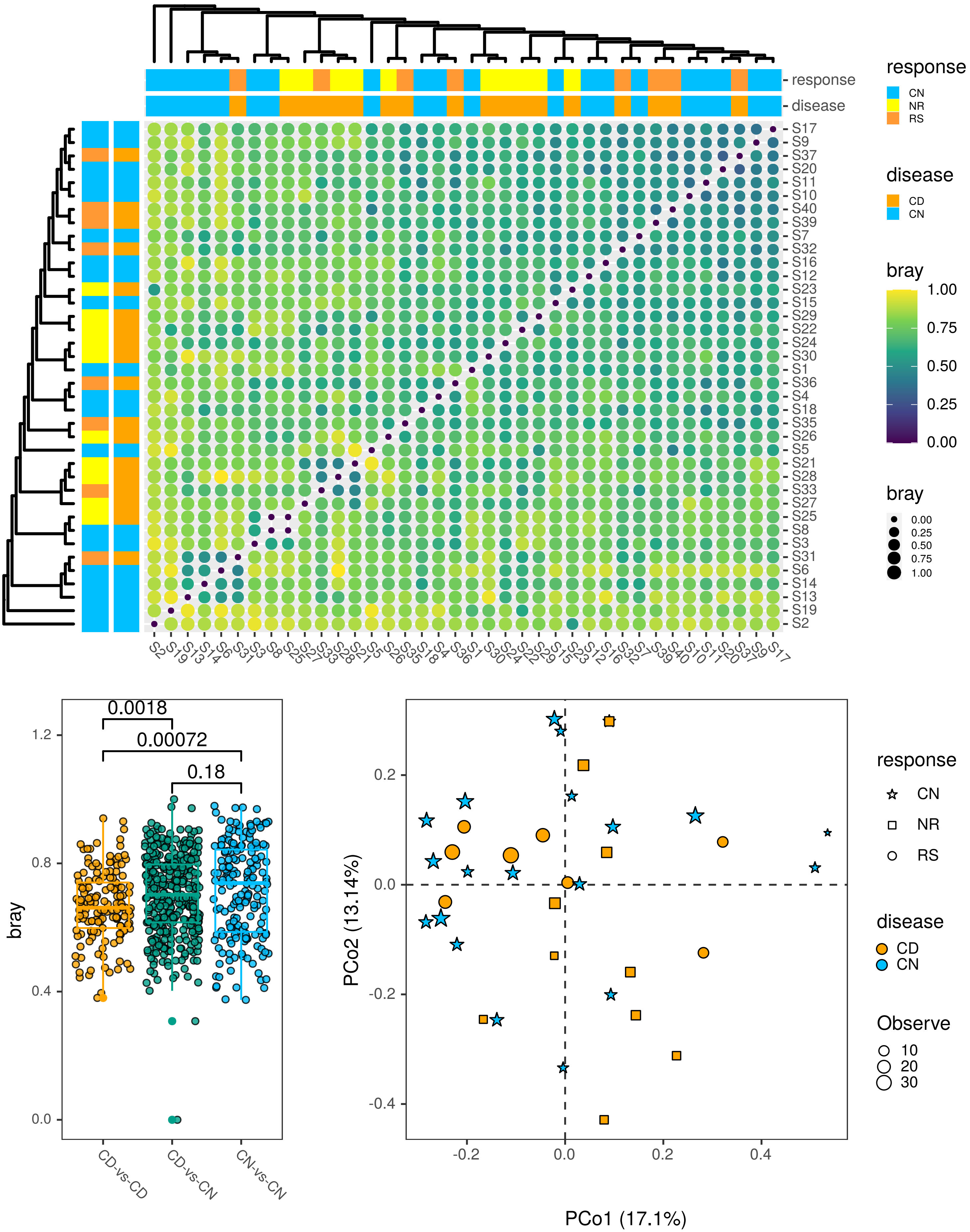


Fig. S18: **The distance heatmap and boxplot and the PCoA plot based on MGS data**

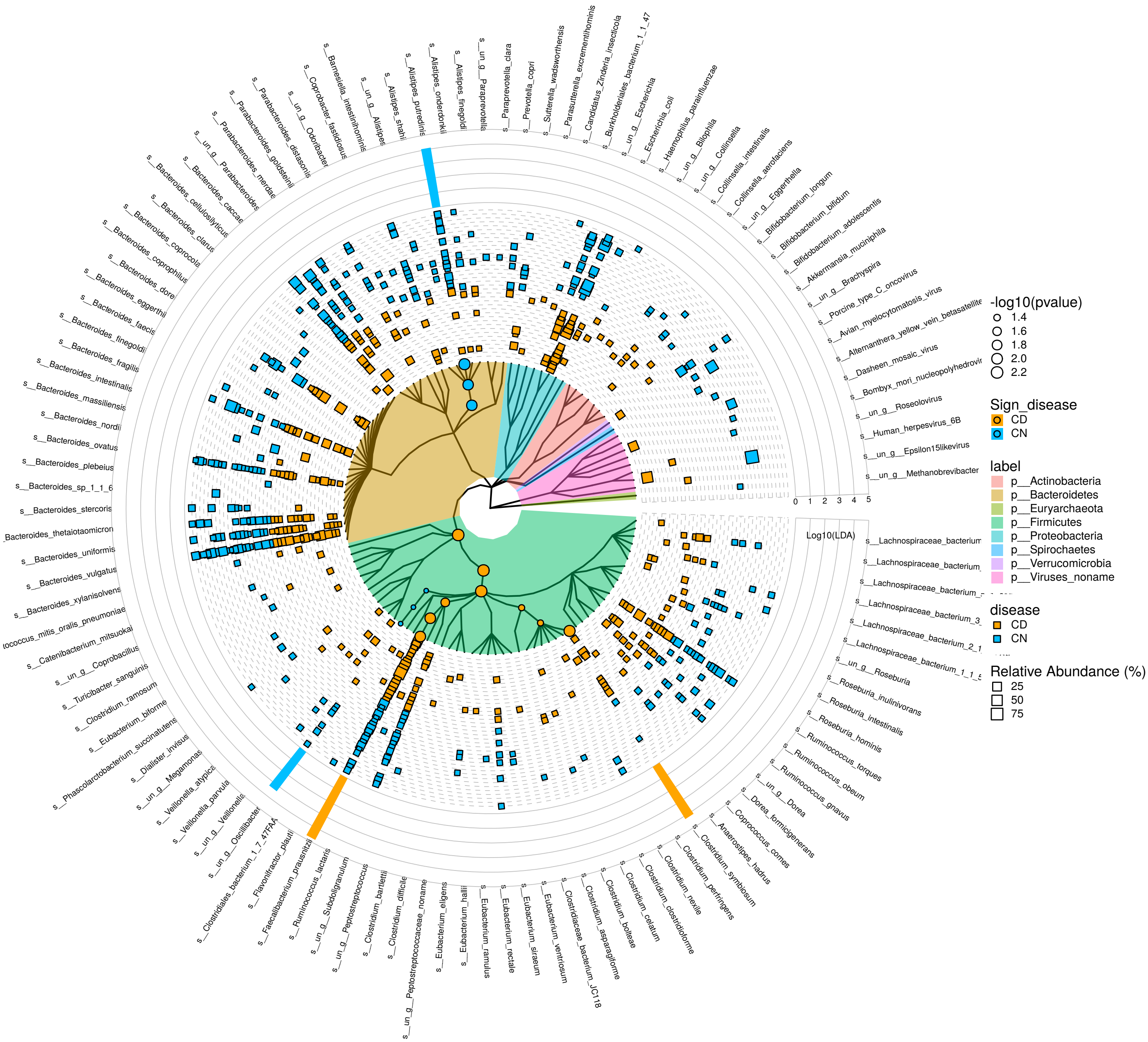


Fig. S19: **The result of different analysis based on MGS data**

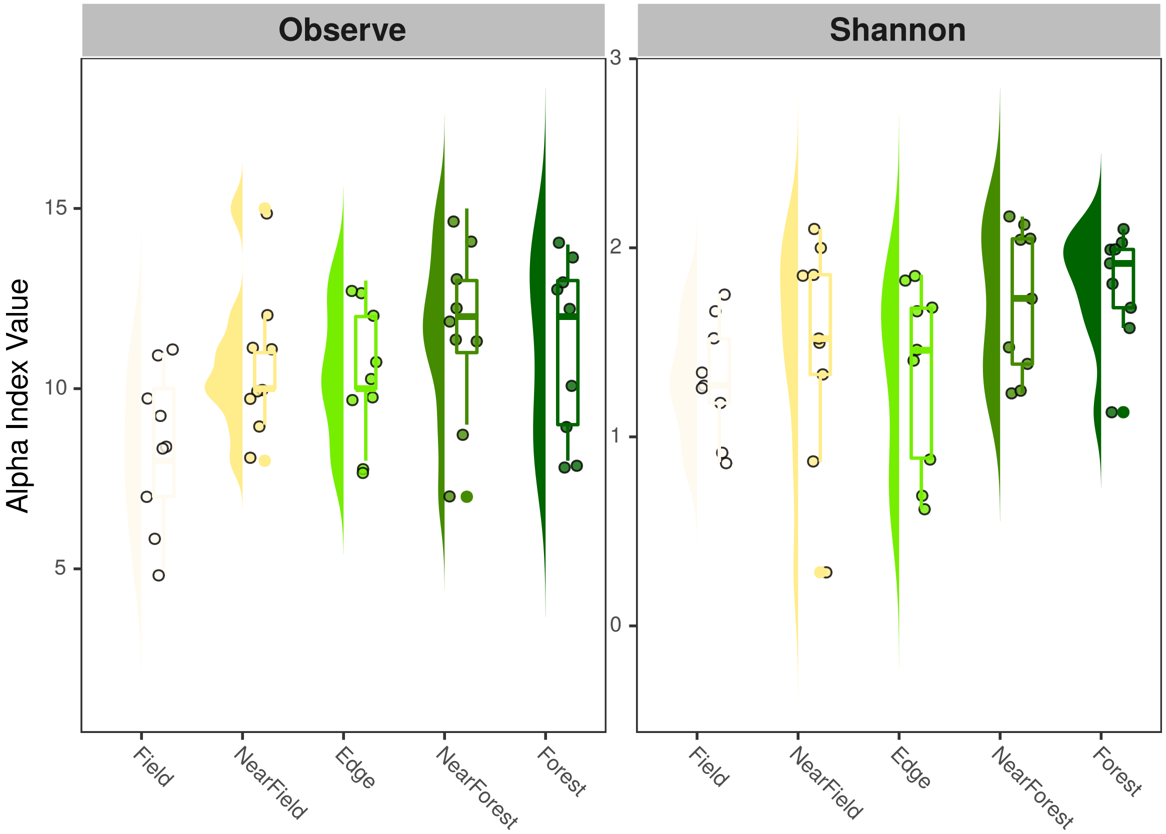


Fig. S20: **The raincloud plot of the alpha diversity of the Mosquito ecology community.** The result of the alpha diversity analysis about the Mosquito ecology study showed that the Mosquito species richness gradually increases from field to forest (field --> near field --> edge --> near field --> forest).

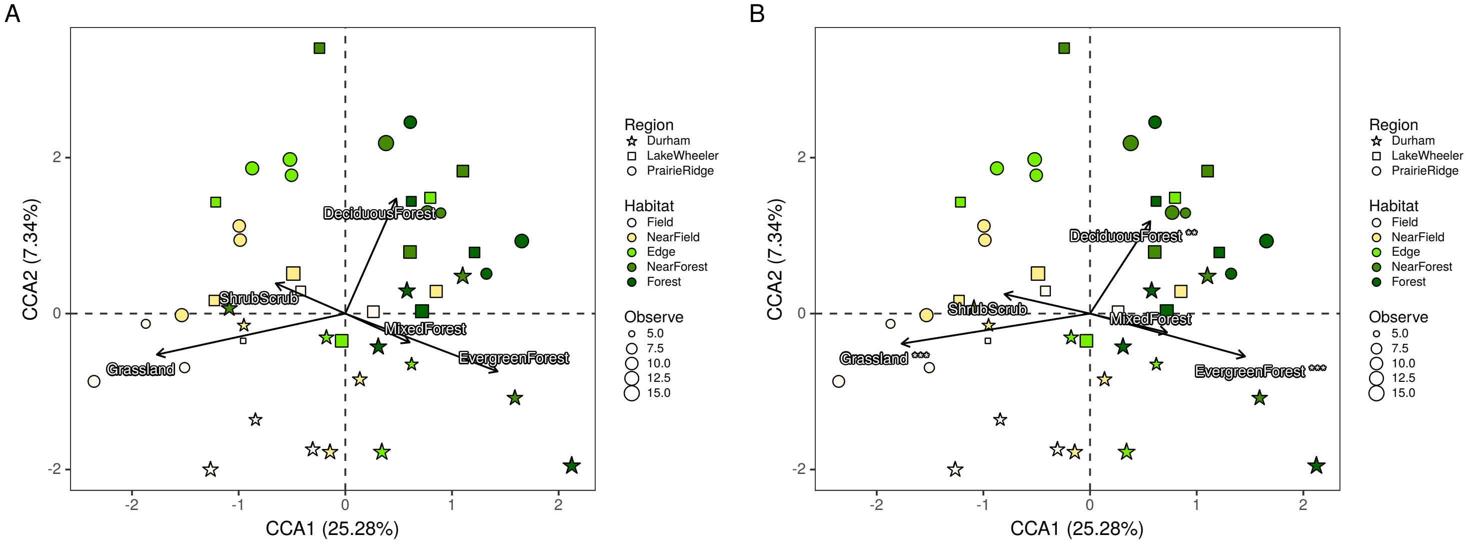


Fig. S21: **The CCA plot of the Mosquito ecology study**. Each point represents one sample, the size of the points represents the observe species of the corresponding sample, the color of the points represents the habitat of the corresponding sample, the shape of points represents the Region of the corresponding sample. And the arrows represent the environment factors, the marked ones by star represent significant related to the Mosquito communities in the study (* 0.05, ** 0.01, *** 0.001).