Supplemental File of

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

Shuangbin Xu, Wenli Tang, Li Zhan, Zehan Dai, Lang Zhou, Tingze Feng, Shanshan Liu, Meijun Chen, 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, obtained from the Integrative Human Microbiome Project Consortium (iHMP) ("The Integrative Human Microbiome Project: Dynamic Analysis of Microbiome-Host Omics Profiles During Periods of Human Health and Disease" 2014).

1.1 Importing the output of dada2

The datasets were obtained from `https://www.microbiomeanalyst.ca/MicrobiomeAnalyst/resources/data/ibd_data.zip`. 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~
```

```
##  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
```

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

1.2 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.

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.

```
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")
```

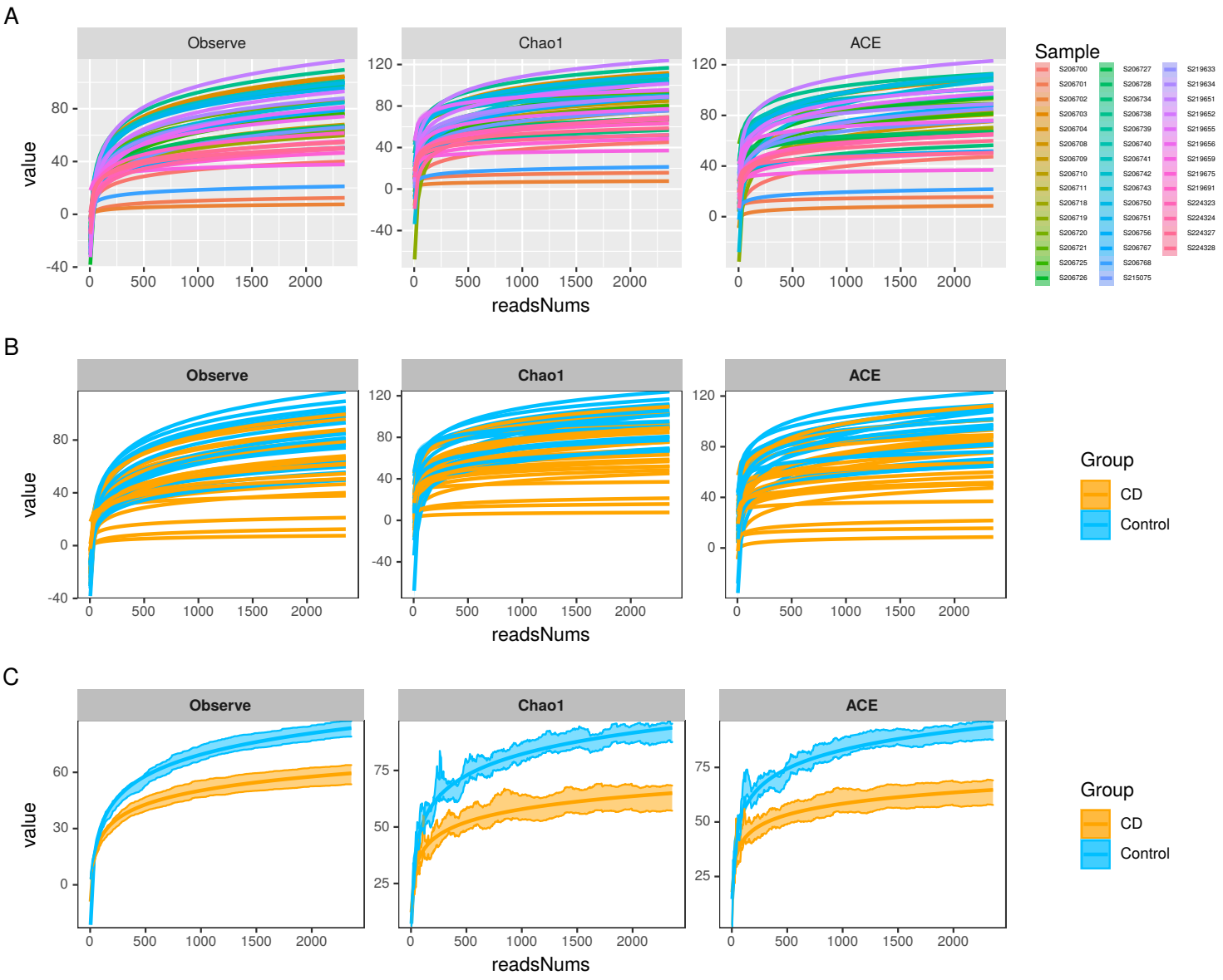


Fig. S1: **This exampls show MicrobiotaProcess provided mp_cal_rarecurve and mp_plot_rarefraction curve to calculate and visualize the rarecurve.** The horizontal coordinate represents the sequencing depth of samples, the vertical coordinate shows the Alpha diversity index (such as Observe OTU, Chao1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

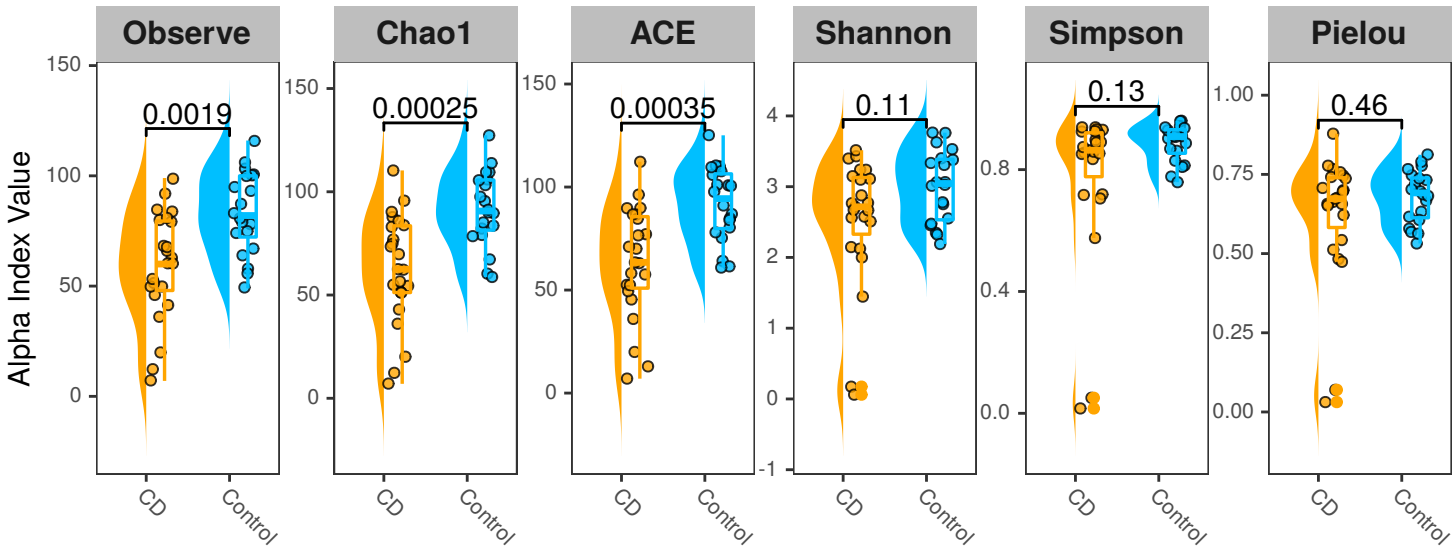


Fig. S2: **The raincloud plot of alpha diversity index** The horizontal coordinate represents each group (by *.group* argument of *mp_plot_alpha*), the vertical coordinate represents the alpha diversity index.

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
```

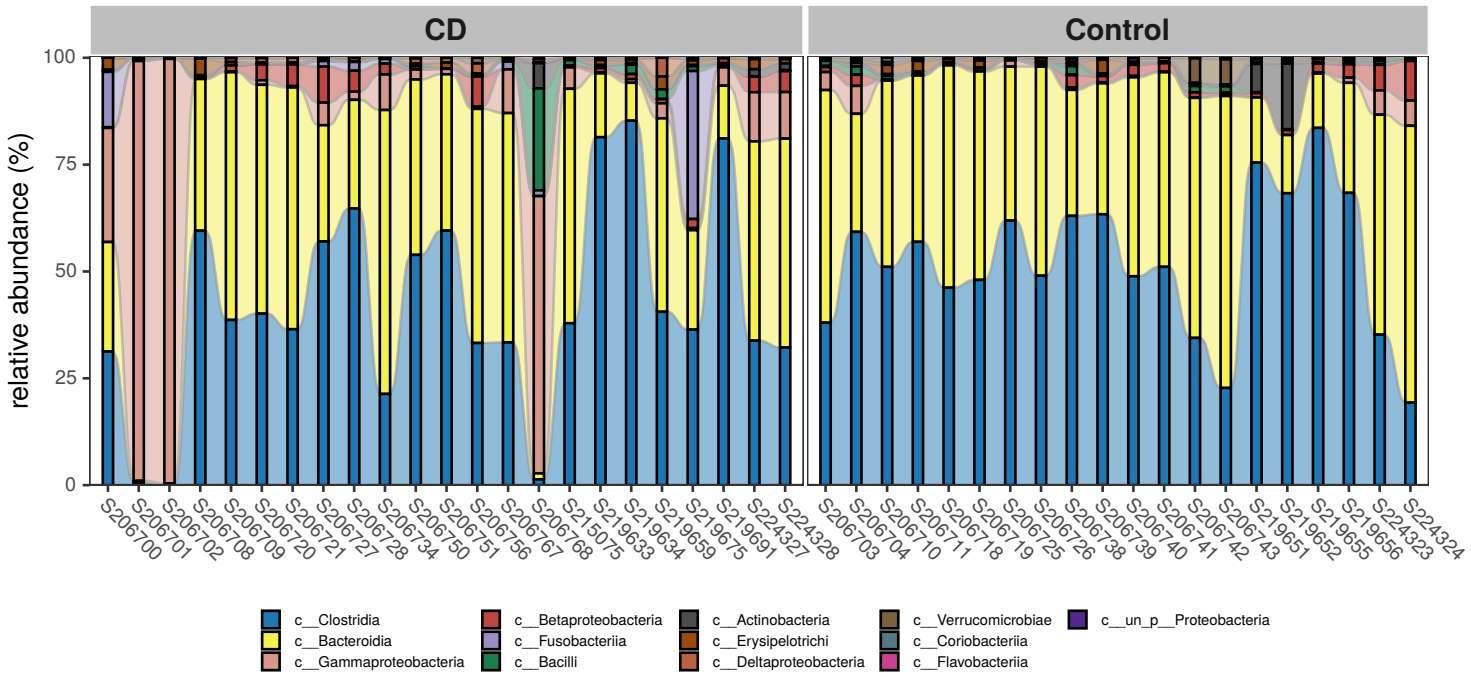


Fig. S3: **The relative abundance of each sample in *class* level**

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")
          #theme(
```

```
          #       legend.position = "right",
          #       legend.key.width = unit(0.6, "cm"),
          #       legend.key.height = unit(0.6, "cm"),
          #       legend.text = element_text(size=8)
          #)

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")
```

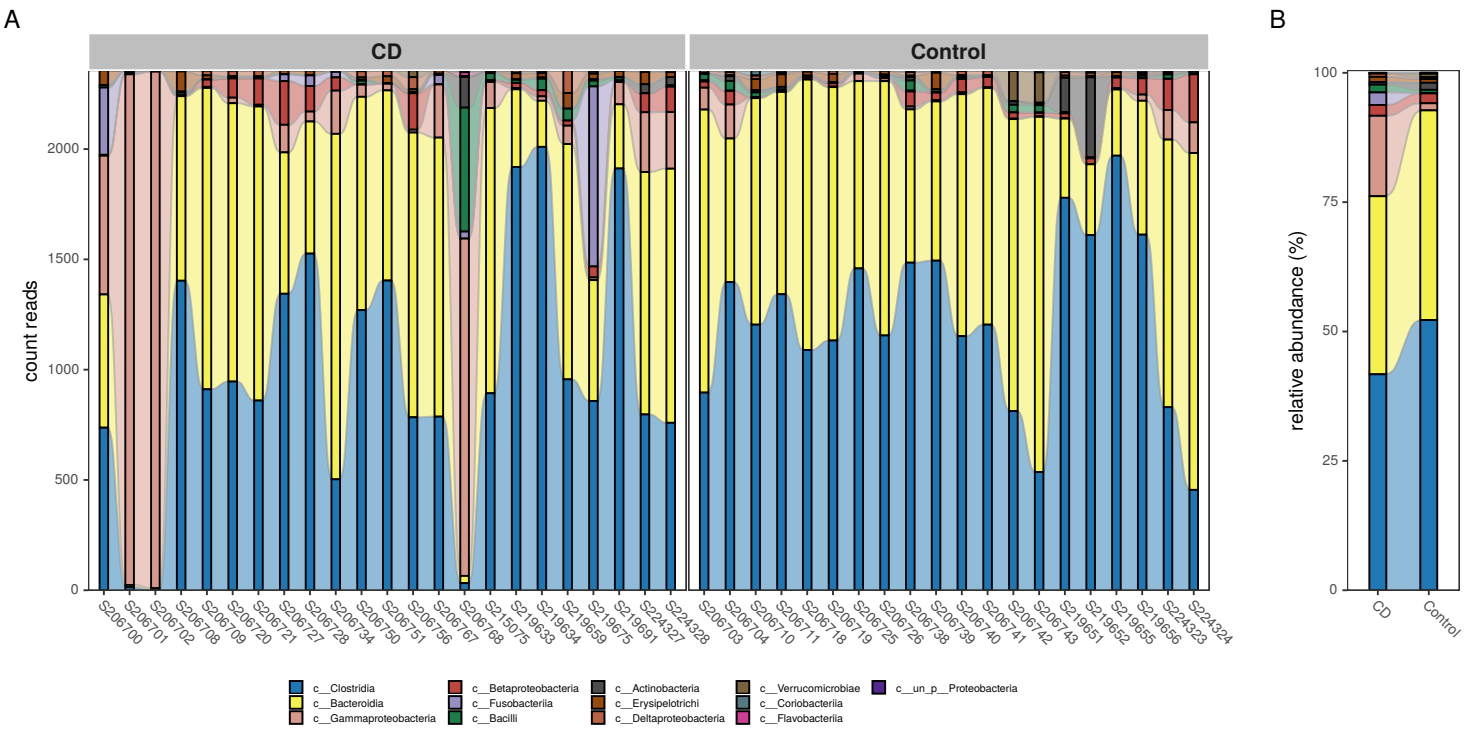


Fig. S4: **This example show how to displayed the abundance (count or other) of sample and the relative abundance of groups**. The Abundance (count by rarefied) of each sample (A) and the relative abundance of group (B), these results show the *Gammaproteobacteria* of *CD* group might be more abundant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

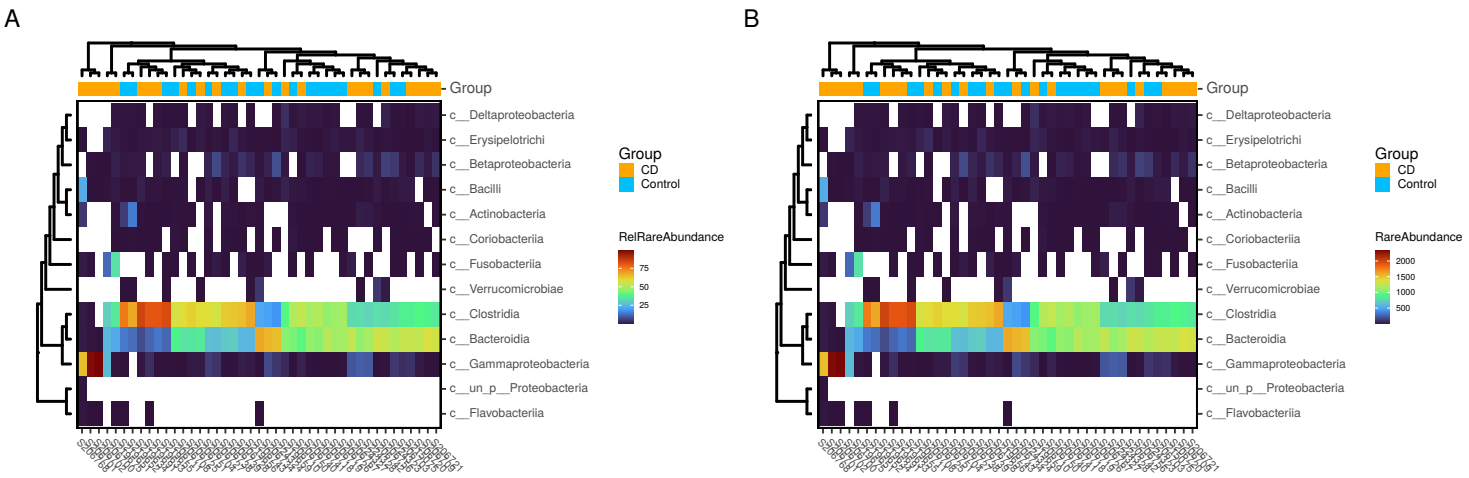


Fig. S5: **The heatmap of abundance for each sample in *class* level.** The color (continuous) of heatmap represents the abundance of taxon, the color of bar represents the group name of sample, the horizontal coordinate represents the sample, and the vertical coordinate represents the taxon.

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
     ) +
     scale_colour_manual(values = cols) +
     scale_fill_viridis_c()
venn_p
```

```
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)
    ) +
```

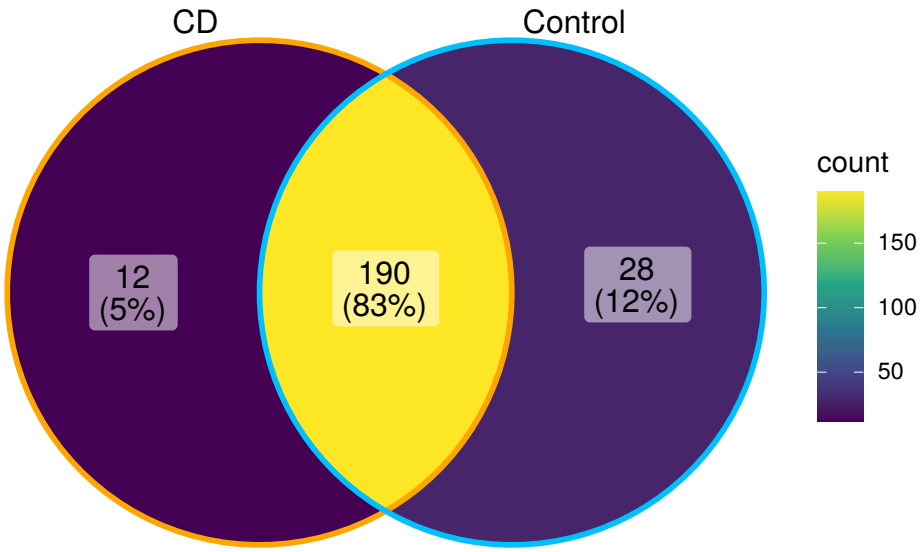


Fig. S6: **The venn diagram for groups in OTU/ASV level**

```
    ggupset::theme_combmatrix(
      combmatrix.label.extra_spacing = 40
    )

upset_p
```

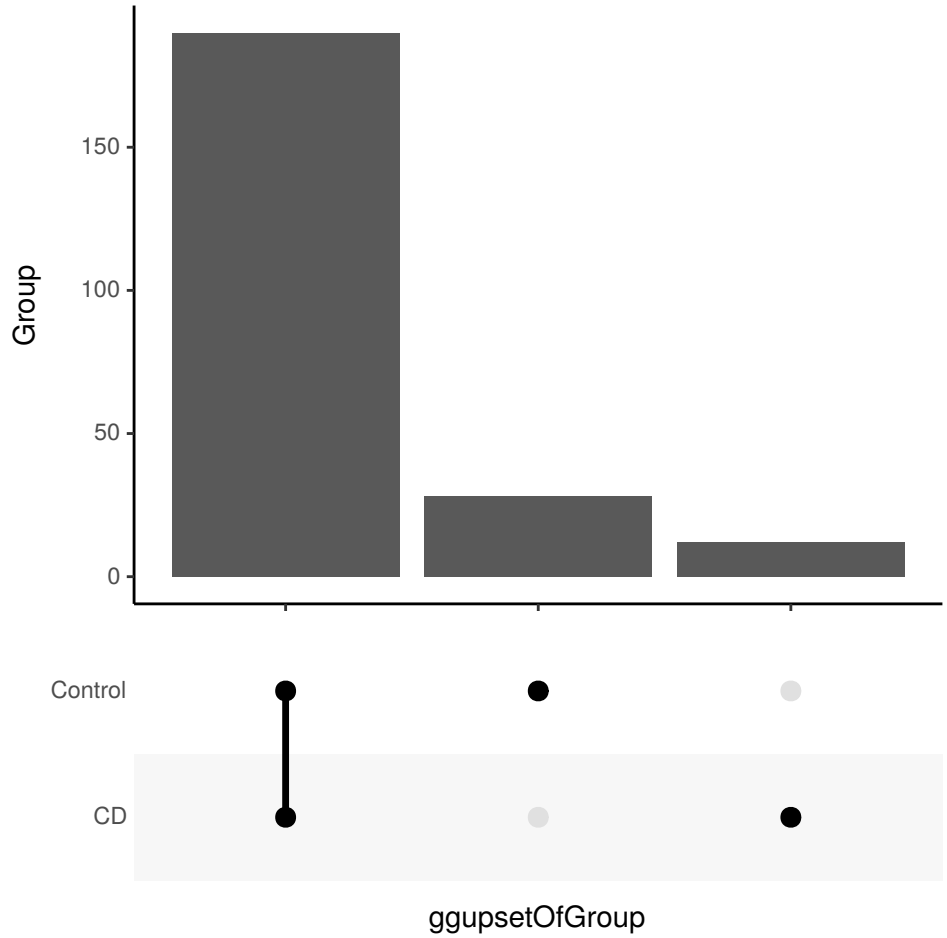


Fig. S7: **The upset plot of groups in OTU/ASV level**

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")
```

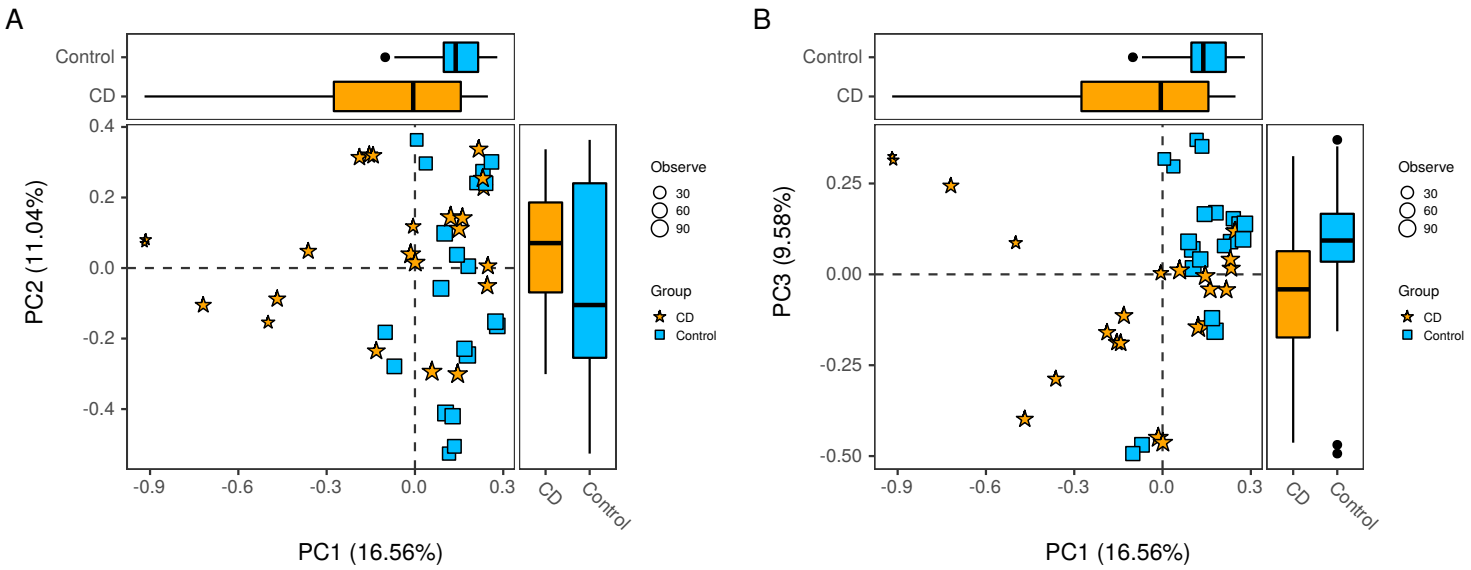


Fig. S8: **The PCA plot of the community**. Each point represents one sample, the size of point represents the observe OTU of the sample. The color of point represents the group name of the sample, based on the first and second component (A), based on the first and third component (B).

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,
               .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")
```

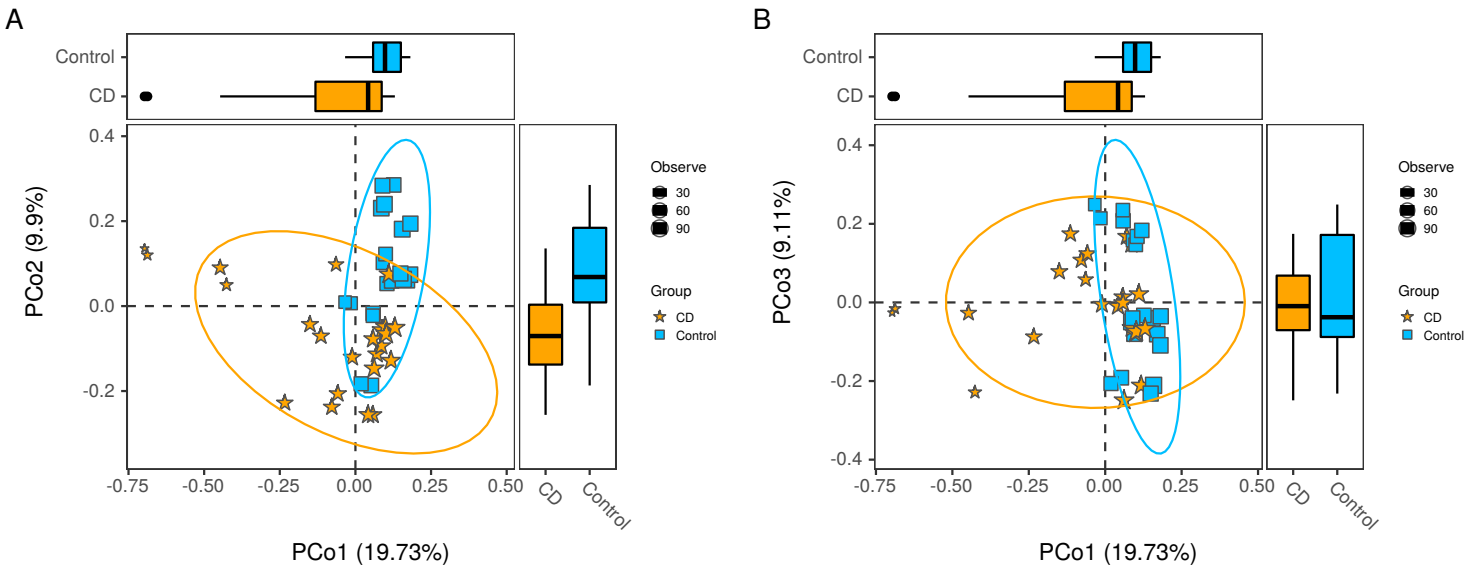


Fig. S9: **The PCoA plot based on Bray-Curtis distance.**

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
          )
```

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

```
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")
```

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
```

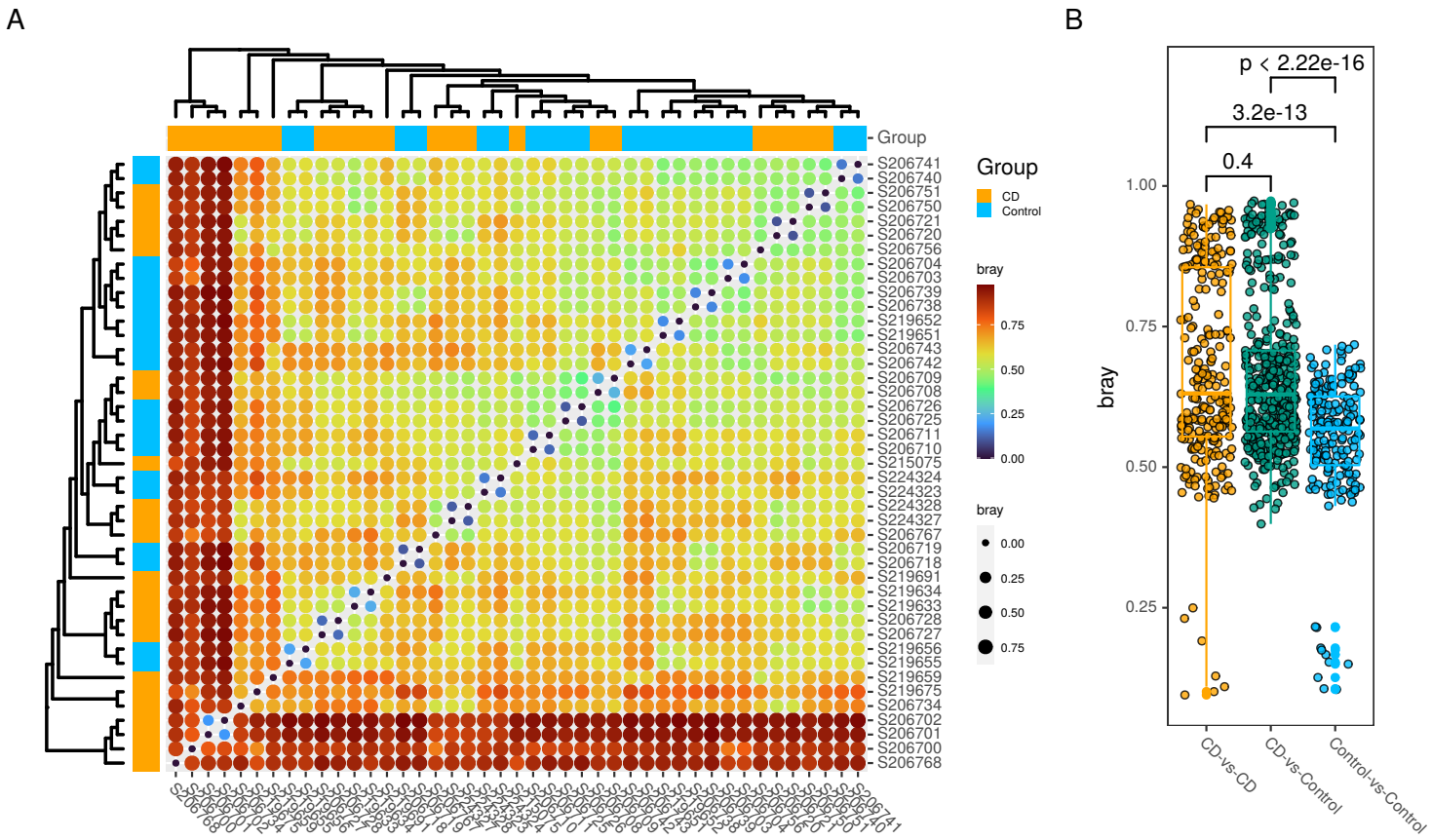


Fig. S10: **The distance heatmap and the boxplot for each sample**. The size and color of the heatmap represent the distance of each sample, the color of bar represent the group of sample (A). The boxplot represent the distance pairs of sample among the group, it show the dissimilarity of sample between the *control* and *CD* is significant, which is consistent with the result of Permutational Multivariate Analysis of Variance 1.5.3.

```
##   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)
```

```r
# 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
```

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).

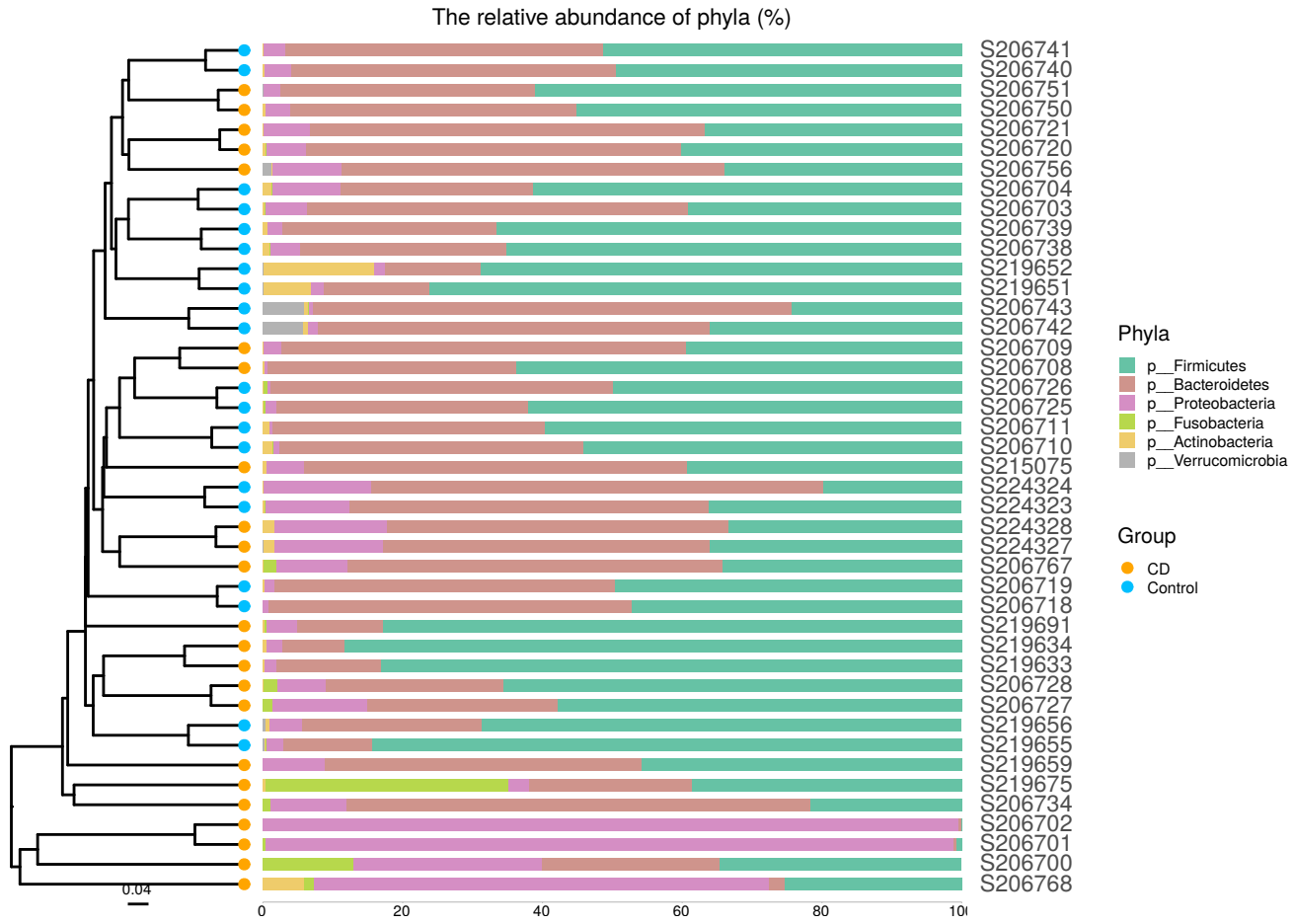


Fig. S11: **The hierarchical clustering plot of samples based on Bray-Curtis distance calculated with abundance of OTU/ASV and the relative Abundance of phyla for samples**

```
# 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
```

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(
```

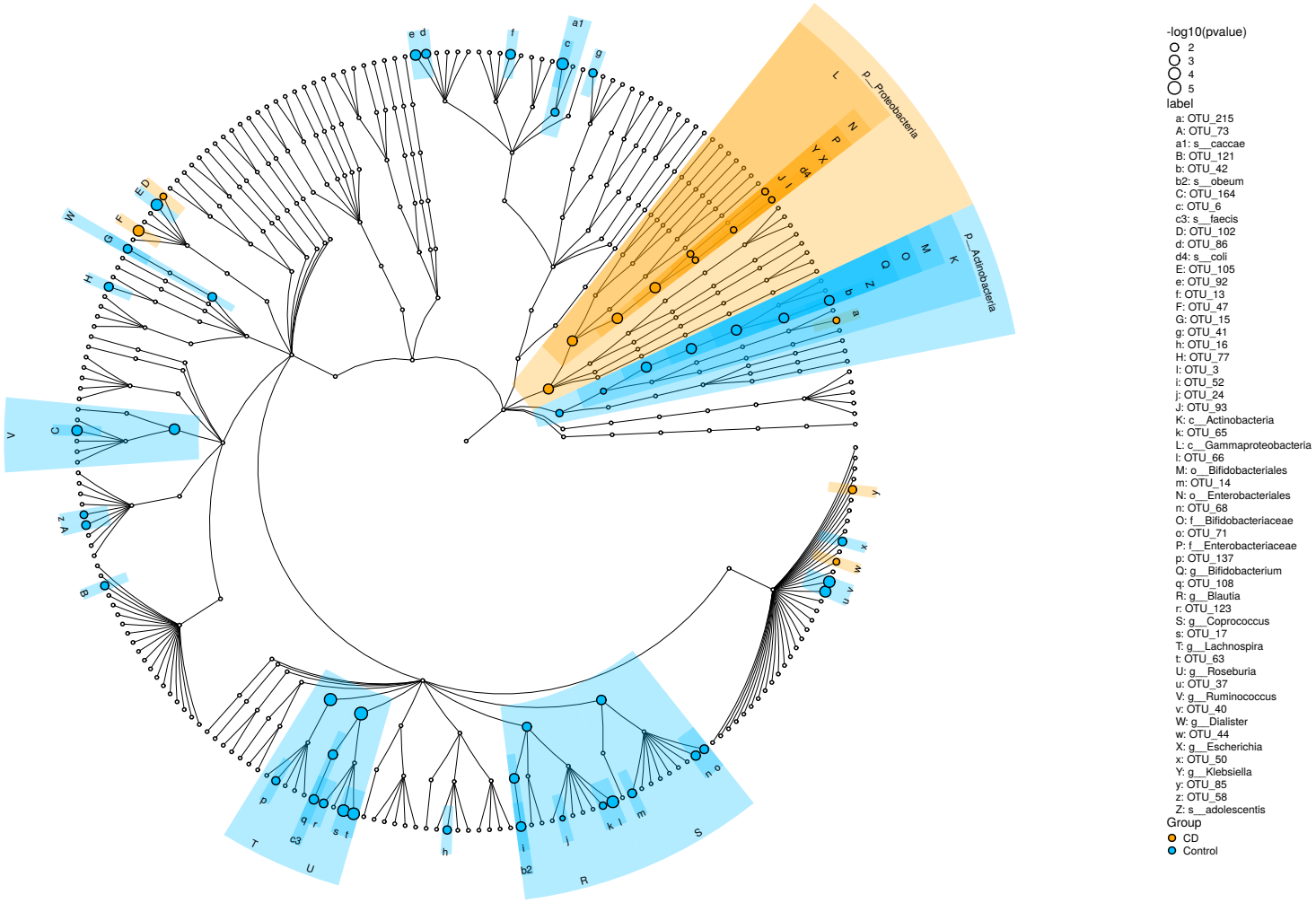


Fig. S12: **The taxa tree clade plot of different analysis result.**

```
        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
```

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

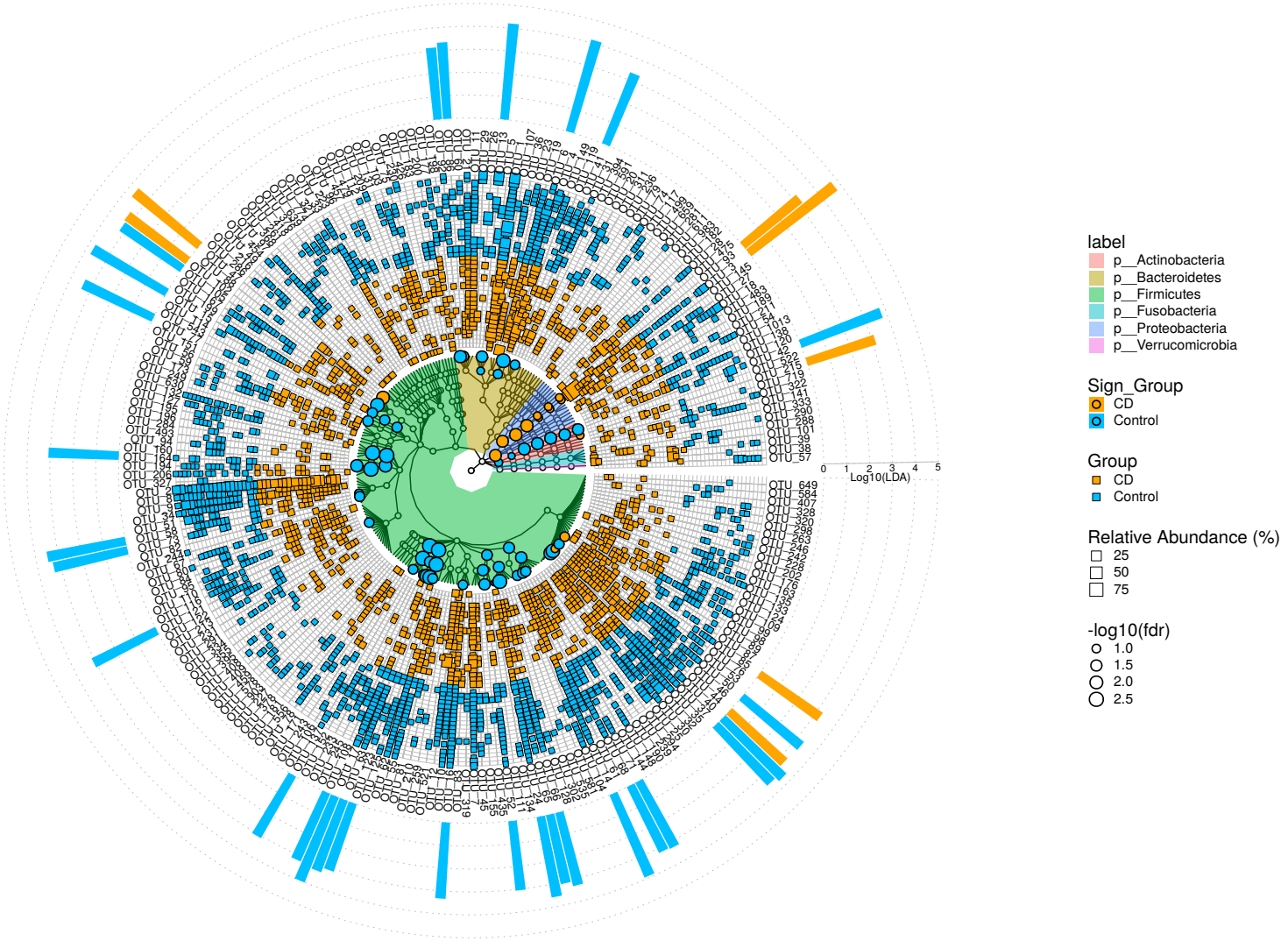


Fig. S13: **The taxa tree of the community with the relative abundance of each OTU/ASV on sample and the LDA of different OTU/ASV**. The taxa tree is built with the taxa of all samples. The high light color of taxa tree represents the phyla of the clade. The external point layer represents the relative 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*.

biomarker is more fluctuant, owing to the influence of samples number.

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

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.

```
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.

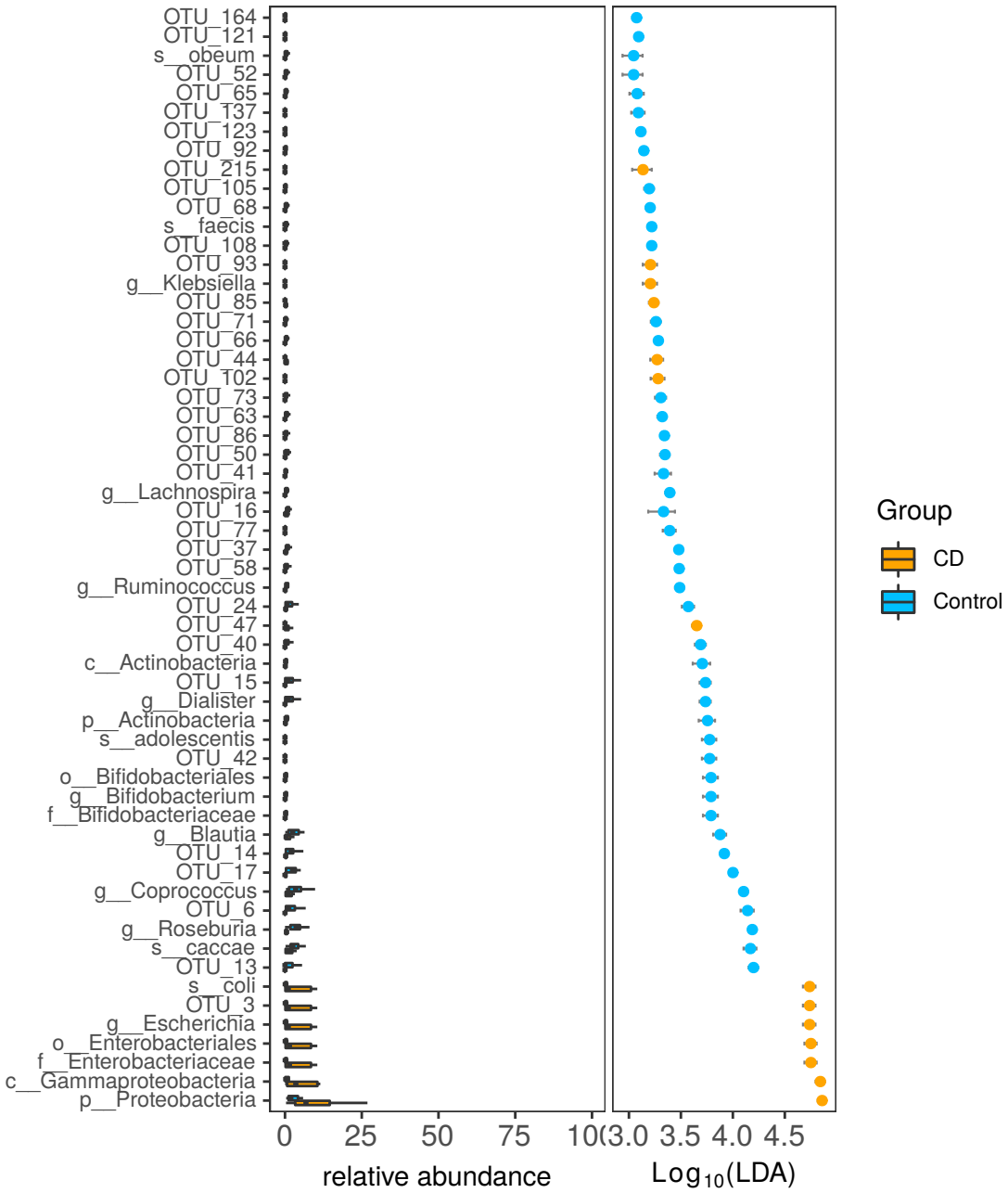


Fig. S14: **The boxplot and the LDA score of different taxa.** The left panel represents the relative abundance of the different taxa, the right panel represents the LDA of different taxa.

```
es_p <- ggeffectsize(obj=deres,
                     lineheight=0.1,
                     linewidth=0.3) +
        scale_color_manual(values=c(cols))

es_p
```

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

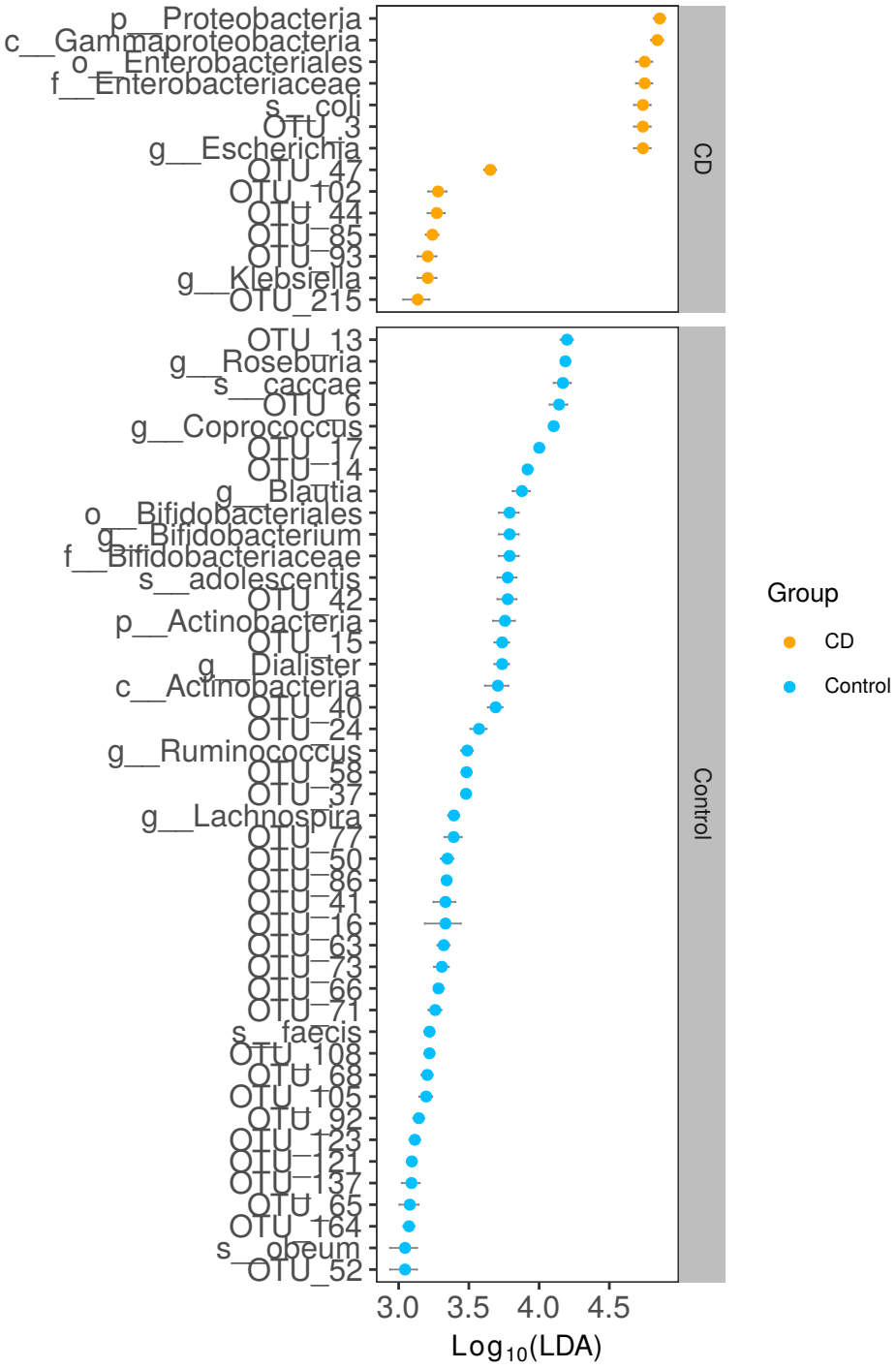


Fig. S15: **The effect size plot of biomarkers**

MicrobiotaProcess to do the related analysis. The datasets were obtained from the github[1].

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")
```

[1] https://github.com/LangilleLab/CD_RF_microbiome

```
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>
```

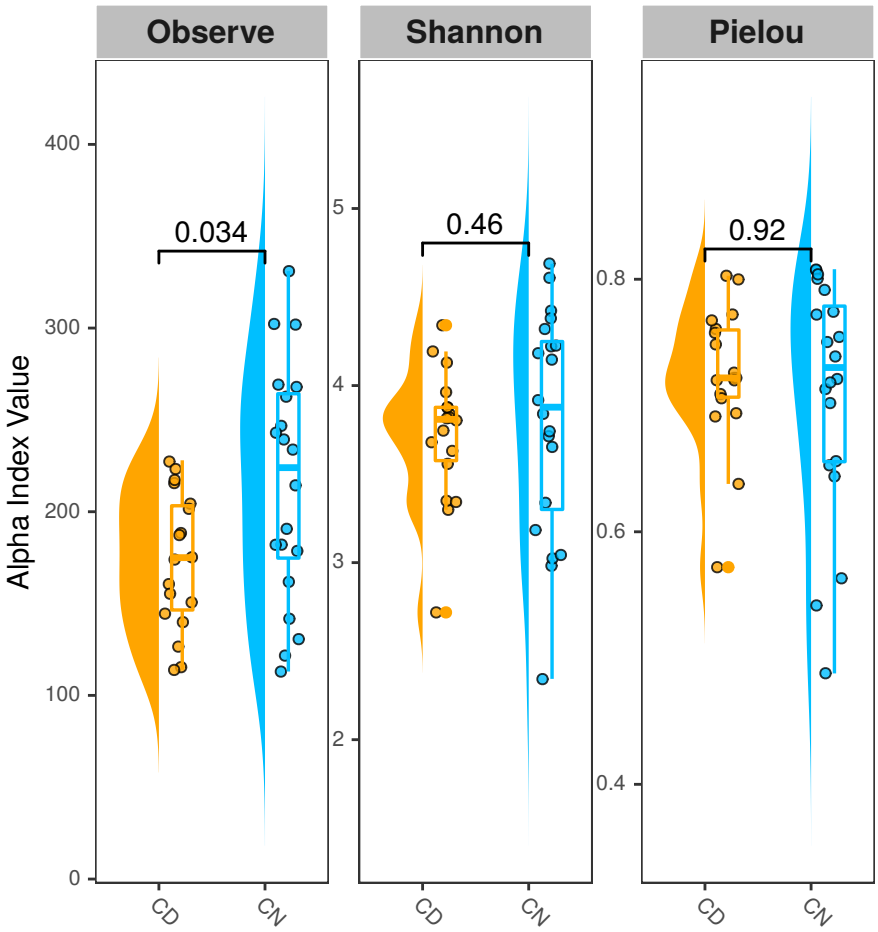
2.1.1 Alpha diversity analysis in 16s level

```
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")
```

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

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

```
p
```



2.1.2 Beta diversity analysis in 16s level

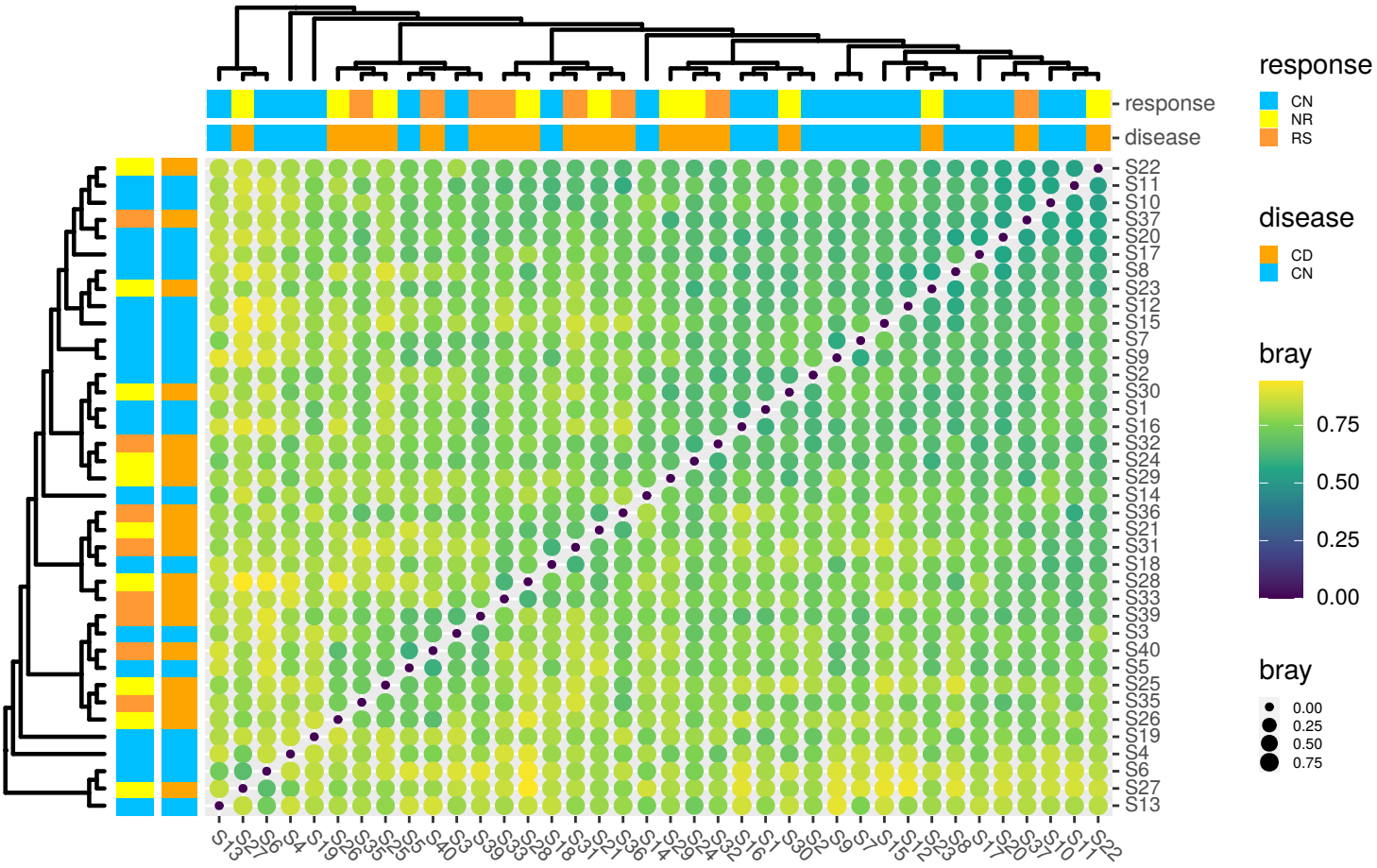
```
mpse16s %<>%
    mp_decostand(
      .abundance = Abundance,
      method = "hellinger"
    )

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(
```

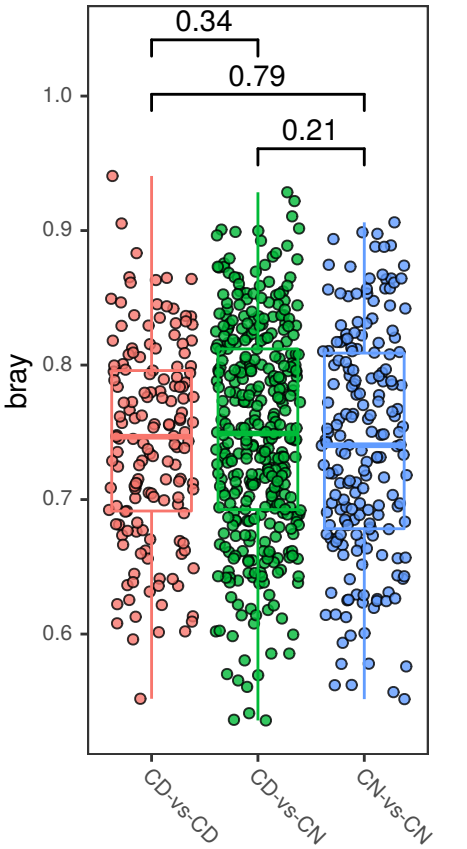
```
                        keywidth = 0.5, keyheight = 0.5,
                        label.theme=element_text(size=6)
                )
            ),
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = scale_fill_manual(
              values=cols2,
              guide = guide_legend(
                        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
      )

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

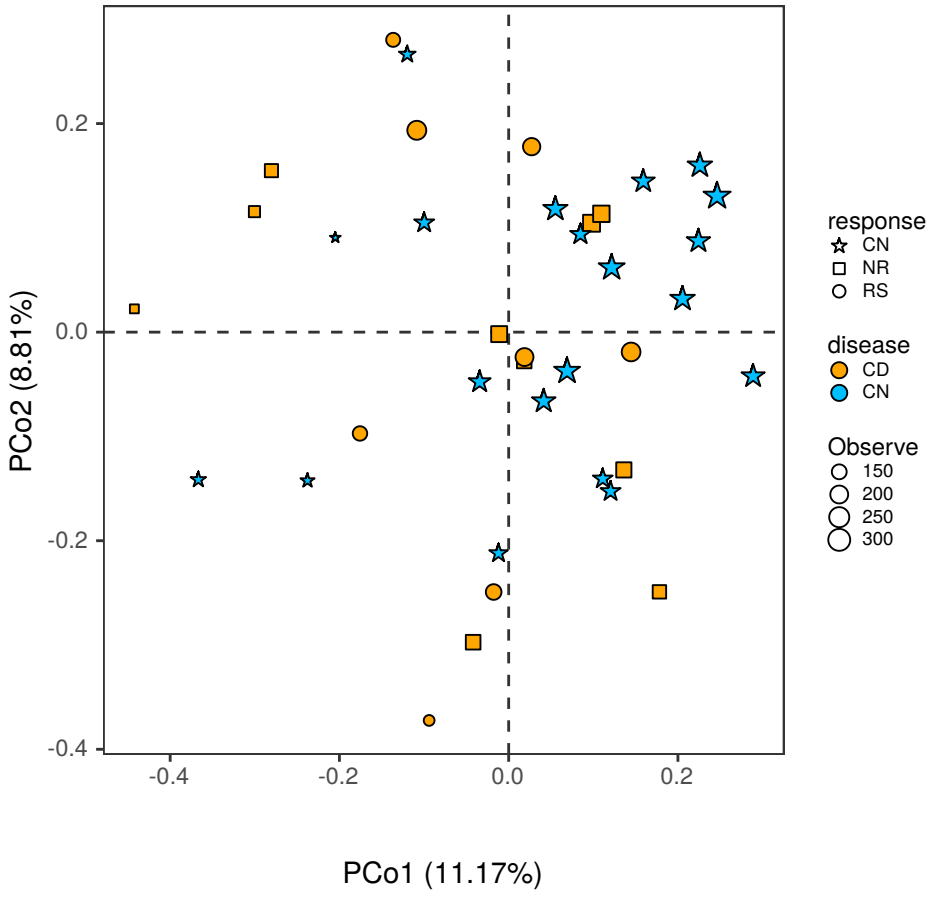
```
##
## Call:
## vegan::adonis(formula = .formula, data = sampleda, permutations = permutations,      method = distmethod)
##
## Permutation: free
## Number of permutations: 9999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## disease    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
```

```
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),
      )

aplot::plot_list(p1, (aplot::plot_list(p2, p3, nrow=1, widths=c(1, 2))), ncol = 1)
```


response
CN
NR
RS
disease
CD
CN
bray
0.75
0.50
0.25
0.00
response
disease
S22
S11
S10
S37
S20
S17
S8
S23
S12
S15
S7
S9
S2
S30
S1
S16
S32
S24
S29
S14
S36
S21
S31
S18
S28
S33
S39
S3
S40
S5
S25
S35
S26
S19
S4
S6
S27
S13


0.34
0.79
0.21
1.0
0.9
0.8
0.7
0.6
bray
CD-vs-CD
CD-vs-CN
CN-vs-CN


0.2
0.0
-0.2
-0.4
PCo2 (8.81%)
-0.4
-0.2
0.0
0.2
PCo1 (11.17%)
response
CN
NR
RS
disease
CD
CN
Observe
150
200
250
300

2.1.3 Composition of the taxonomy in 16s level

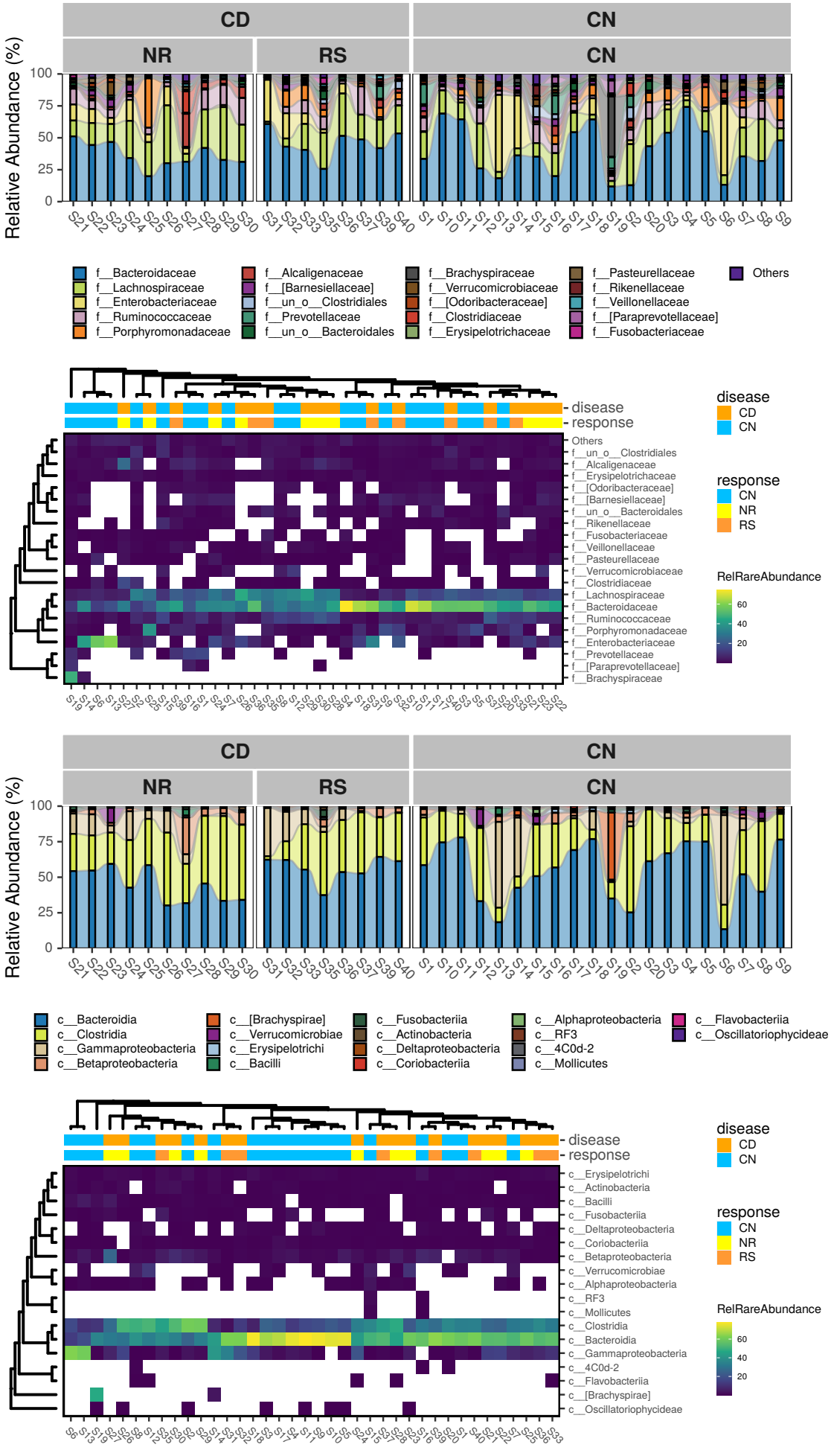
```
mpse16s %<>%
    mp_cal_abundance(
       .abundance=RareAbundance
    )
```

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

```
p1 <- mpse16s %>%
      mp_plot_abundance(
        .abundance = RareAbundance,
        taxa.class = Family,
        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 = Family,
        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),
```

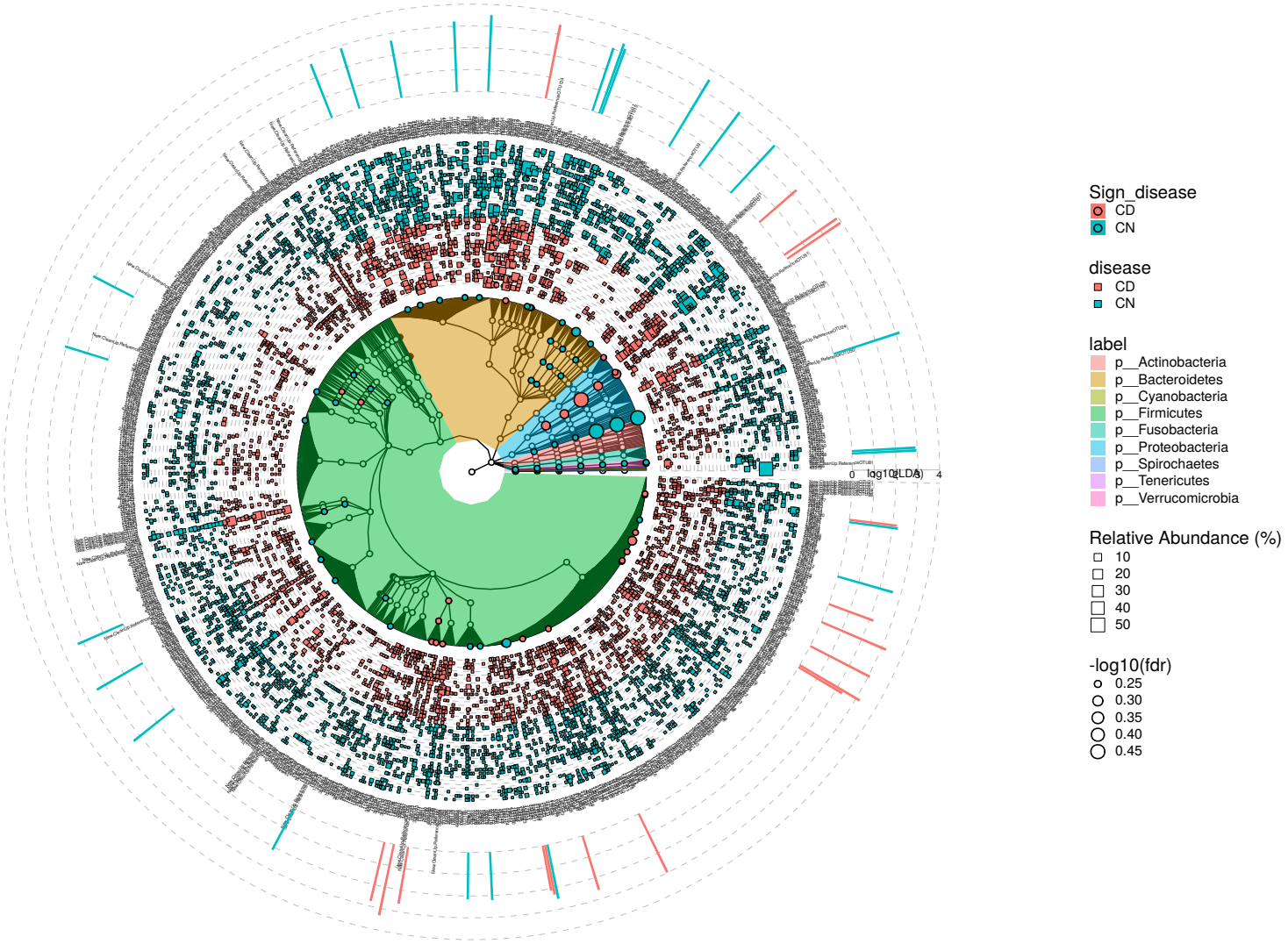
```
        aes_var = disease
      ) %>%
      set_scale_theme(
        x = theme(
          legend.key.height = unit(0.3, "cm"),
          legend.key.width = unit(0.3, "cm"),
          legend.spacing.y = unit(0.02, "cm"),
          legend.text = element_text(size = 7),
          legend.title = element_text(size = 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
f__Bacteroidaceae
f__Lachnospiraceae
f__Enterobacteriaceae
f__Ruminococcaceae
f__Porphyromonadaceae
f__Alcaligenaceae
f__[Barnesiellaceae]
f__un_o__Clostridiales
f__Prevotellaceae
f__un_o__Bacteroidales
f__Brachyspiraceae
f__Verrucomicrobiaceae
f__[Odoribacteraceae]
f__Clostridiaceae
f__Erysipelotrichaceae
f__Pasteurellaceae
f__Rikenellaceae
f__Veillonellaceae
f__[Paraprevotellaceae]
f__Fusobacteriaceae
Others
disease
response
CD
CN
NR
RS
RelRareAbundance
60
40
20
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

2.1.4 Different analysis in 16S level

```
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)
```

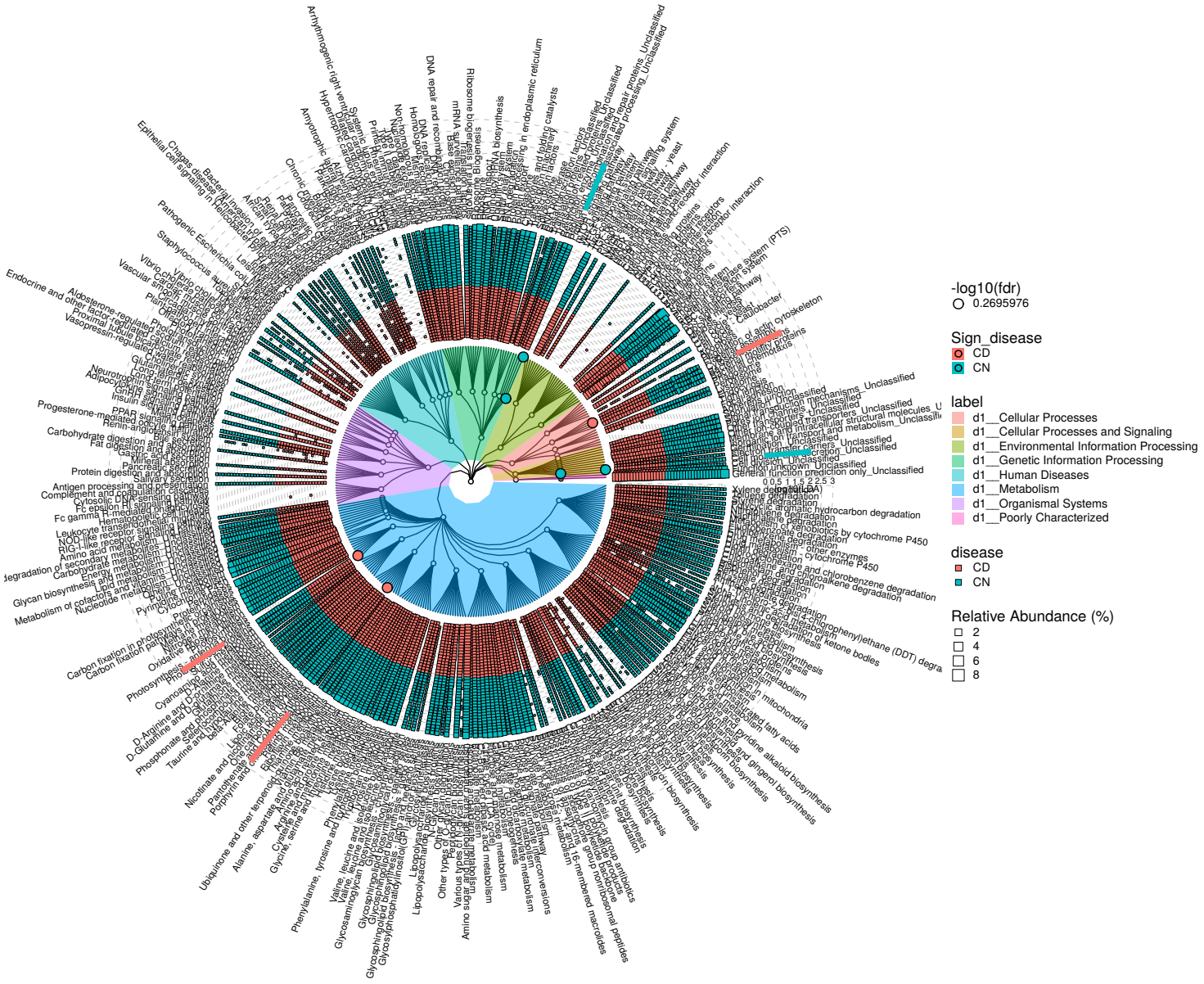


Fig. S16: **The result of the different analysis based on the KEGG pathway data**

```
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")
```

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

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

```
p
```

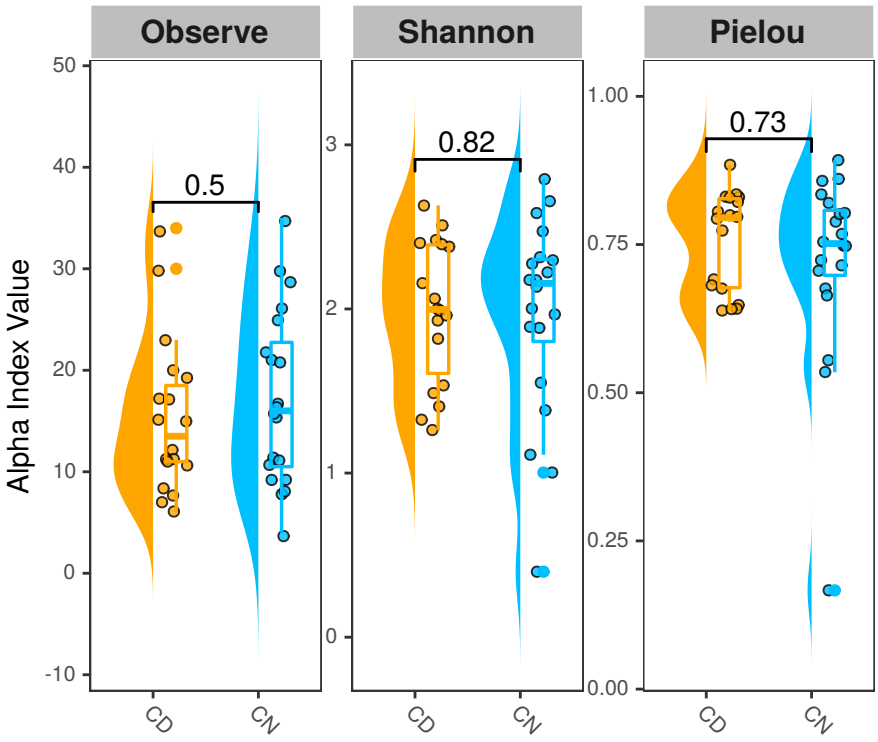


Fig. S17: **The alpha diversity boxplot based on MGS data**

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
      )

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)
```

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>
```

```
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[2].

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[3]

```
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 | Taxanomy=NULL
##    OTU      Sample Abundance Region Transect Habitat DeciduousForest
##    <chr>    <chr>      <int> <chr>  <chr>    <chr>             <dbl>
```

[2] https://github.com/rgriff23/Mosquito_ecology
[3] http://www.randigriffin.com/2017/05/23/mosquito-community-ecology-in-vegan.html

```
##  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>
```

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

```
cols = terrain.colors(6)[5:1]
# 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>
```

```
# 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
```

```
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
```

```
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+
           BarrenLand+
           Building+
           Pavement+
           CultivatedCrops+
           TrapNights
    )
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 (25.64%) <dbl>, CCA2 (7.29%) <dbl>, CCA3 (5.26%) <dbl>
```

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

```
## The object contained internal attribute: CCA
```

```
## Call: cca(formula = x ~ DeciduousForest + EvergreenForest + Grassland +
## MixedForest + ShrubScrub + BarrenLand + Building + Pavement +
## CultivatedCrops + TrapNights, data = sampleda)
##
##               Inertia Proportion Rank
## Total          1.1595     1.0000
## Constrained    0.5666     0.4886   10
## Unconstrained  0.5929     0.5114   22
## Inertia is scaled Chi-square
##
## Eigenvalues for constrained axes:
##    CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7    CCA8    CCA9   CCA10
## 0.29734 0.08452 0.06096 0.04522 0.03015 0.02045 0.01379 0.00798 0.00370 0.00248
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.11841 0.08877 0.06437 0.05667 0.03812 0.03365 0.02996 0.02830
## (Showing 8 of 22 unconstrained eigenvalues)
```

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

```
## The object contained internal attribute: CCA
```

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

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

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

```
## The object contained internal attribute: CCA CCA_ENVFIT
```

```
##
## ***VECTORS
##
##                     CCA1     CCA2     CCA3     r2 Pr(>r)
## DeciduousForest  0.34344  0.69184  0.63514 0.3703 0.0022 **
## EvergreenForest  0.96073 -0.26084 -0.09462 0.6400 0.0001 ***
## Grassland       -0.97657 -0.21082 -0.04316 0.7647 0.0001 ***
## MixedForest      0.88963 -0.07882 -0.44982 0.2276 0.0505 .
## ShrubScrub      -0.86471 -0.00015  0.50227 0.2278 0.0565 .
```

```
## BarrenLand      -0.88243  0.17676 -0.43597 0.1097 0.2953
## Building        -0.22535  0.59166 -0.77405 0.1816 0.1019
## Pavement        -0.58707  0.65575  0.47470 0.0031 0.9844
## CultivatedCrops -0.37129  0.52212 -0.76781 0.2267 0.0415 *
## TrapNights       0.52580  0.81643  0.23870 0.1941 0.0925 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 9999
```

```
# visualization
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),
     )
p
```

4 Session information

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

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.4 LTS
##
## Matrix products: default
## BLAS:   /mnt/d/UbuntuApps/R/4.1.1/lib/R/lib/libRblas.so
## LAPACK: /mnt/d/UbuntuApps/R/4.1.1/lib/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
```

```
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggstar_1.0.3                patchwork_1.1.1
##  [3] ggtreeExtra_1.5.1           ggtree_3.3.0.900
##  [5] forcats_0.5.1               ggnewscale_0.4.5
##  [7] coin_1.4-2                  survival_3.2-13
##  [9] ggplot2_3.3.5               MicrobiotaProcess_1.7.3.990
## [11] SummarizedExperiment_1.24.0 Biobase_2.54.0
## [13] GenomicRanges_1.46.0        GenomeInfoDb_1.30.0
## [15] IRanges_2.28.0              S4Vectors_0.32.0
## [17] BiocGenerics_0.40.0         MatrixGenerics_1.6.0
## [19] matrixStats_0.61.0          kableExtra_1.3.4
##
## loaded via a namespace (and not attached):
##   [1] ggVennDiagram_1.1.4    TH.data_1.1-0          colorspace_2.0-2
##   [4] ggsignif_0.6.3         class_7.3-19           ellipsis_0.3.2
##   [7] modeltools_0.2-23      XVector_0.34.0         aplot_0.1.1
##  [10] proxy_0.4-26           rstudioapi_0.13        farver_2.1.0
##  [13] ggrepel_0.9.1          fansi_0.5.0            mvtnorm_1.1-3
##  [16] xml2_1.3.2             codetools_0.2-18       splines_4.1.1
##  [19] ggh4x_0.2.0            libcoin_1.0-9          knitr_1.36
##  [22] jsonlite_1.7.2         cluster_2.1.2          compiler_4.1.1
##  [25] httr_1.4.2             assertthat_0.2.1       Matrix_1.3-4
##  [28] fastmap_1.1.0          lazyeval_0.2.2         cli_3.1.0
##  [31] htmltools_0.5.2        tools_4.1.1            gtable_0.3.0
##  [34] glue_1.5.0             GenomeInfoDbData_1.2.7 corrr_0.4.3
##  [37] dplyr_1.0.7            Rcpp_1.0.7             rhdf5filters_1.6.0
##  [40] vctrs_0.3.8            Biostrings_2.62.0      ape_5.5-3
##  [43] svglite_2.0.0          nlme_3.1-153           iterators_1.0.13
##  [46] gghalves_0.1.1         ggalluvial_0.12.3      xfun_0.28
##  [49] stringr_1.4.0          rvest_1.0.2            lifecycle_1.0.1
##  [52] zlibbioc_1.40.0        MASS_7.3-54            zoo_1.8-9
##  [55] scales_1.1.1           biomformat_1.22.0      ggupset_0.3.0
##  [58] parallel_4.1.1         sandwich_3.0-1         rhdf5_2.38.0
##  [61] RColorBrewer_1.1-2     yaml_2.2.1             gridExtra_2.3
##  [64] ggfun_0.0.4            dtplyr_1.1.0           yulab.utils_0.0.4.901
##  [67] stringi_1.7.5          foreach_1.5.1          e1071_1.7-9
##  [70] tidytree_0.3.6         permute_0.9-5          ggside_0.1.2
##  [73] rlang_0.4.12           pkgconfig_2.0.3        systemfonts_1.0.3
##  [76] bitops_1.0-7           evaluate_0.14          lattice_0.20-45
##  [79] Rhdf5lib_1.16.0        sf_1.0-3               purrr_0.3.4
##  [82] treeio_1.18.0          labeling_0.4.2         tidyselect_1.1.1
##  [85] plyr_1.8.6             magrittr_2.0.1         bookdown_0.24
##  [88] R6_2.5.1               generics_0.1.1         multcomp_1.4-17
##  [91] DelayedArray_0.20.0    DBI_1.1.1              pillar_1.6.4
##  [94] withr_2.4.2            mgcv_1.8-38            units_0.7-2
##  [97] RCurl_1.98-1.5         tibble_3.1.6           crayon_1.4.2
## [100] KernSmooth_2.23-20     utf8_1.2.2             RVenn_1.1.0
## [103] rmarkdown_2.11         grid_4.1.1             data.table_1.14.2
## [106] vegan_2.5-7            classInt_0.4-3         digest_0.6.28
## [109] webshot_0.5.2          tidyr_1.1.4            gridGraphics_0.5-1
## [112] munsell_0.5.0          viridisLite_0.4.0      ggplotify_0.1.0
```

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.

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.

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.

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

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.

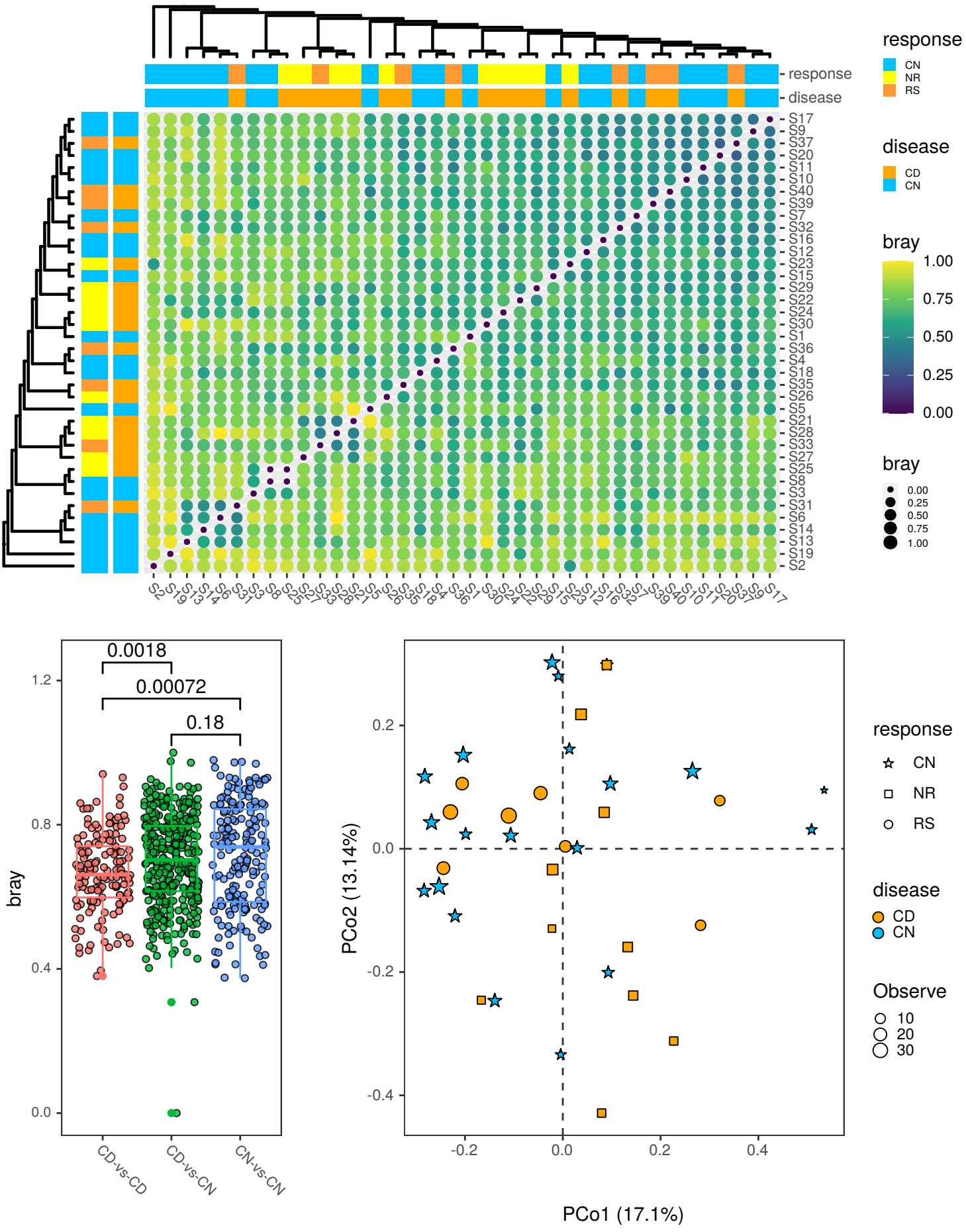


Fig. S18: **The distance heatmap and boxplot and the PCoA plot based on MGS data**

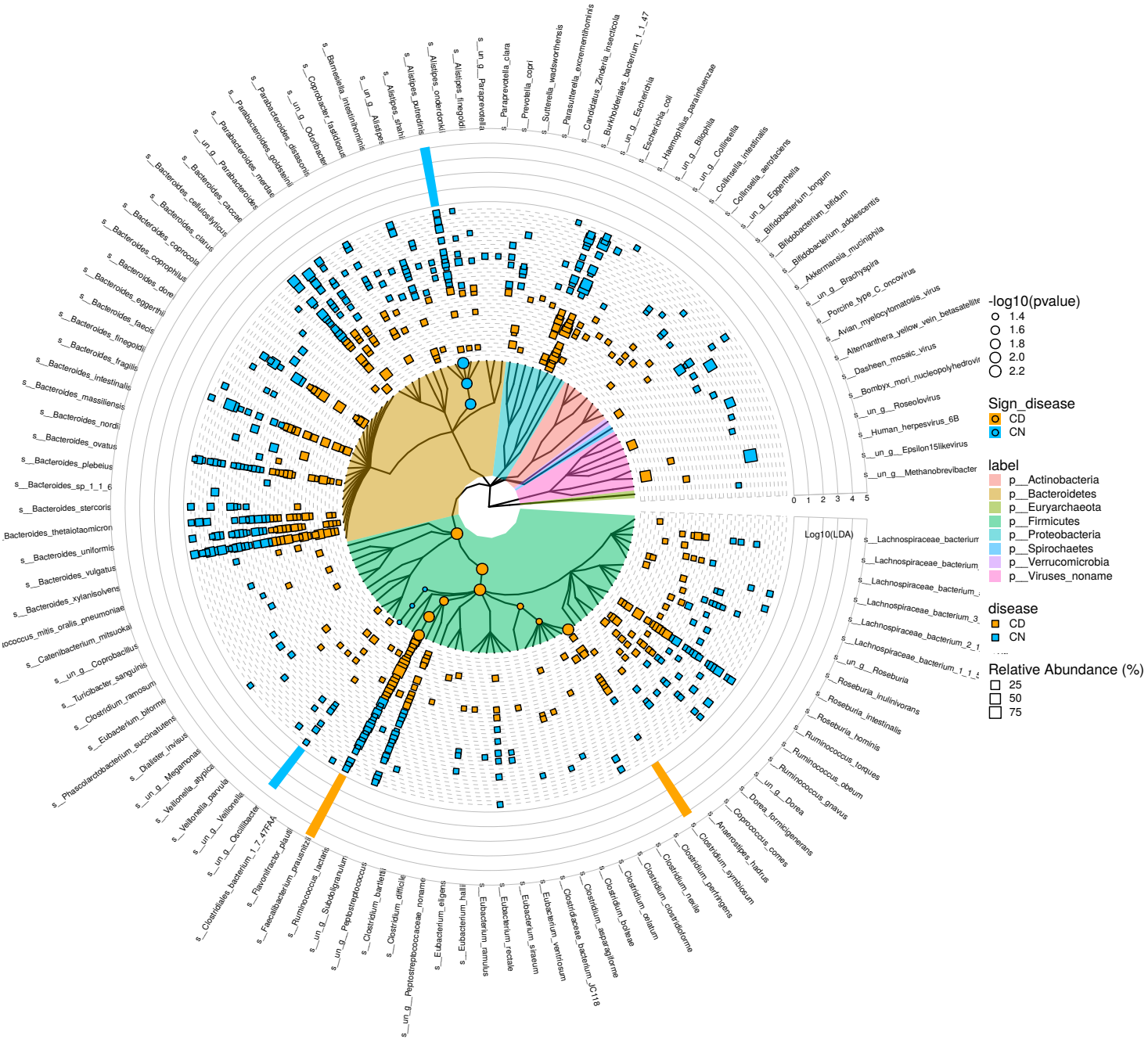


Fig. S19: DiffMGS

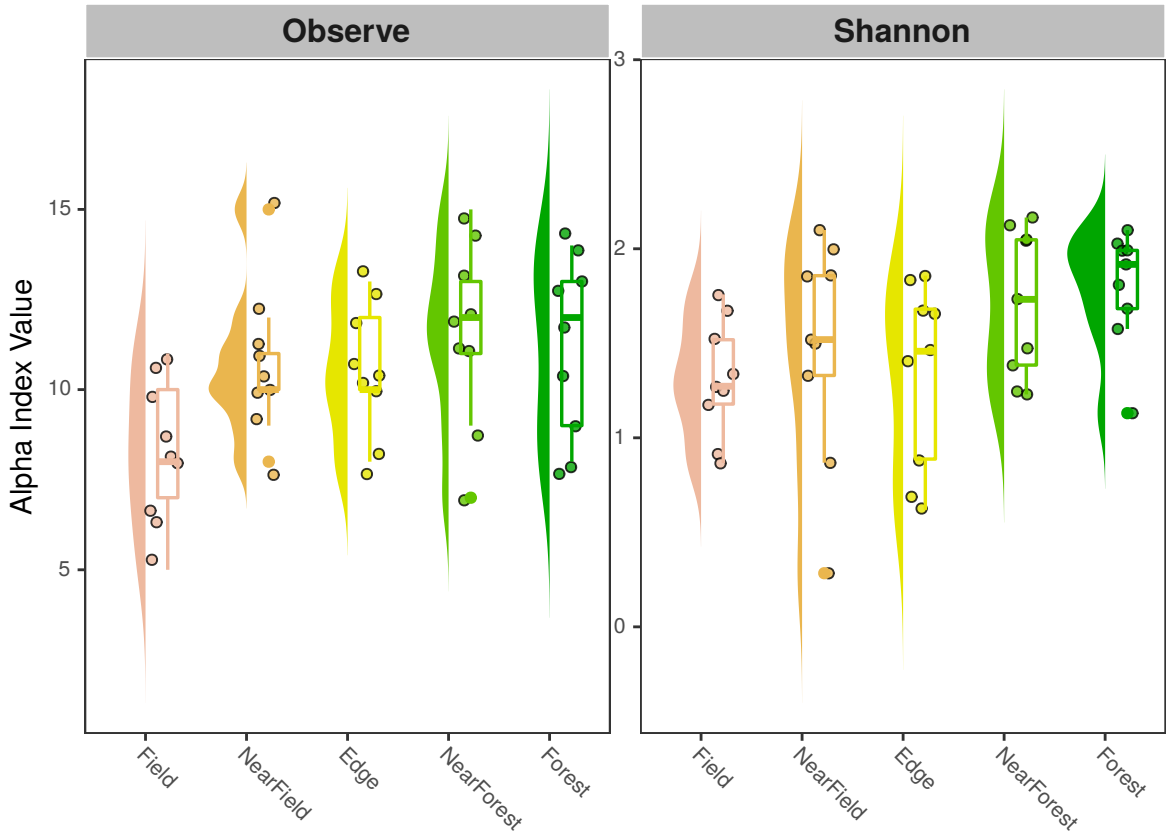


Fig. S20: **The raincloud plot of the alpha diversity of the Mosquito ecology community.** The result of the alpha diversity analysis about the Mosquito ecology study showed that the Mosquito species richness gradually increases from field to forest (field --> near field --> edge --> near field --> forest).

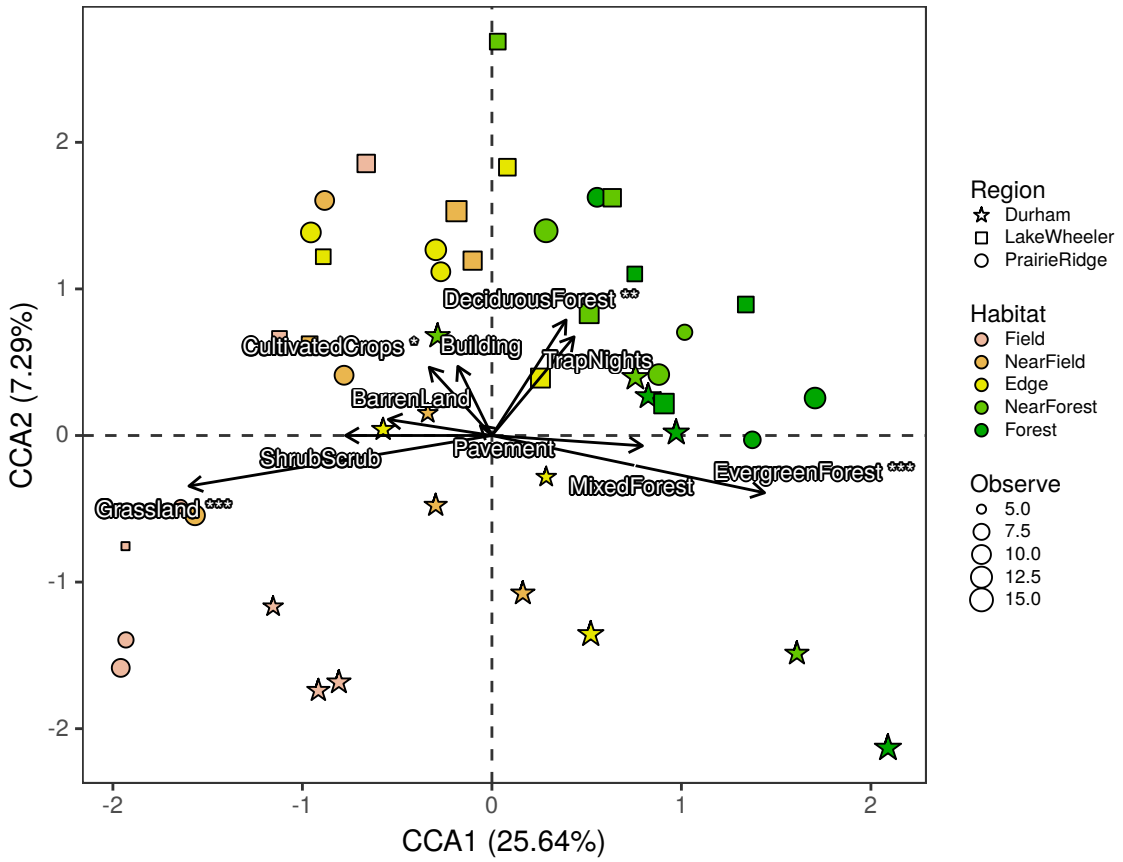


Fig. S21: **The CCA plot of the Mosquito ecology study**. Each point represents one sample, the size of the points represents the observe species of the corresponding sample, the color of the points represents the habitat of the corresponding sample, the shape of points represents the Region of the corresponding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