WASHU Cystic Fibrosis Lung Transplant Microbiome Project

Ahmed A Metwally (ametwa2@uic.edu)

2/24/2019

```
# Load needed libraries and set working directory
library("heatmap3")
library("data.table")
library("ggplot2")
library("phyloseq")
library("DESeq2")
library("vegan")
library("devtools")
library("MetaLonDA")
library("zoo")
library('plyr')
library('reshape2')

## Set seed to ensure reproducibility
set.seed(635473)

# Color palettes:
cbbPalette_9 = c("#56B4E9", "sienna1", "#009E73", "darkorchid",  "#F0E442", "#0072B2",
                 "#CC79A7", "#999999", "#000000")
cbbPalette_10 = c("#999999", "#F0E442", "#0072B2", "#D55E00", "#CC79A7", "#000000",
                  "#E69F00", "#56B4E9", "#009E73", "darkorchid")
cbbPalette_12 = c("#999999", "#F0E442", "#0072B2", "#D55E00", "#CC79A7", "#000000",
                  "#E69F00", "#56B4E9", "#009E73", "darkorchid", "mistyrose1", "sienna1")
```

```
###############################################
######### Prepare PhyloSeq Object ############
###############################################
### Prepare count matrix
countTable = read.csv(file="data/LungTxCF12_countMatrix.csv", header=TRUE,
                      check.names = FALSE, row.names = 1)
countTable = as.matrix(countTable)

## Prepare taxa matrix
taxaTable = read.csv(file="data/LungTxCF12_OTUMatrix.csv", header=TRUE,
                     check.names = FALSE, row.names = 1)
taxaTable = as.matrix(taxaTable)

## Prepare metadata matrix
meta = read.csv(file="data/Supplement_table_1_CF_12_v2.6.csv", header = TRUE)

## Prepare phyloseq data object
OTU = otu_table(countTable, taxa_are_rows = TRUE)
TAX = tax_table(taxaTable)
META = sample_data(meta)
sample_names(META) = meta$SampleIDFP
```

```
physeq = phyloseq(OTU, TAX, META)

## Remove taxa with unannotated superkingdom
physeq = subset_taxa(physeq, Superkingdom != "")


##############################################
############    NORMALIZATION   ##############
##############################################
gm_mean = function(x, na.rm=TRUE){
  exp(sum(log(x[x > 0]), na.rm = na.rm) / length(x))
}
phseq_dds = phyloseq_to_deseq2(physeq, ~ Outcome)
geoMeans = apply(counts(phseq_dds), 1, gm_mean)
phseq_dds_est = estimateSizeFactors(phseq_dds, geoMeans = geoMeans)
otu_matrix_norm = as.data.frame(counts(phseq_dds_est, normalized=TRUE))
OTU_norm = otu_table(otu_matrix_norm, taxa_are_rows = TRUE)
physeq_norm = phyloseq(OTU_norm, TAX, META)
```

```
###########################################################
## Segragate eukaryota & microbial & Bacteria & Virus taxa
###########################################################
## Unnormlaized taxa
eukaryota = subset_taxa(physeq, Superkingdom == "Eukaryota")
microbial = subset_taxa(physeq, Superkingdom != "Eukaryota")
bacteria = subset_taxa(microbial, Superkingdom == "Bacteria")
virus = subset_taxa(microbial, Superkingdom == "Viruses")
archaea = subset_taxa(microbial, Superkingdom = "Archaea")

## Normalized taxa
eukaryota_norm = subset_taxa(physeq_norm, Superkingdom == "Eukaryota")
microbial_norm = subset_taxa(physeq_norm, Superkingdom != "Eukaryota")
bacteria_norm = subset_taxa(microbial_norm, Superkingdom == "Bacteria")
virus_norm = subset_taxa(microbial_norm, Superkingdom == "Viruses")
archaea_norm = subset_taxa(microbial_norm, Superkingdom = "Archaea")


## Extract human reads
human = subset_taxa(physeq, Species == "Homo_sapiens")
human_df = as.data.frame(apply(otu_table(human), 2, sum))
colnames(human_df)[1]="count"
write.csv(human_df, file="Number_human_reads_per_sample.csv")

## List number of taxa per bacterial taxonomic level
length(unique(tax_table(bacteria_norm)[,"Phylum"]))
```

```
## [1] 23
```

```
length(unique(tax_table(bacteria_norm)[,"Family"]))
```

```
## [1] 204
```

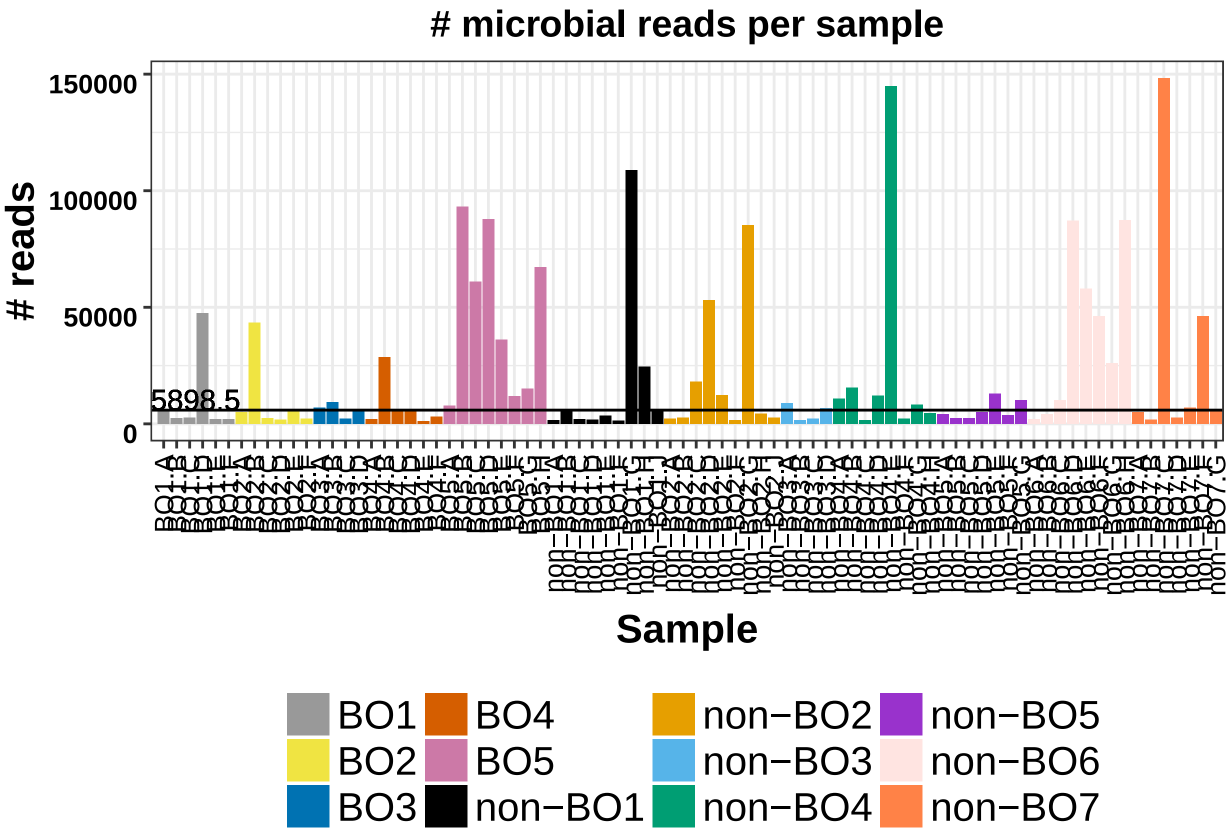
```
length(unique(tax_table(bacteria_norm)[,"Genus"]))
```

```
## [1] 510
```

```r
length(unique(tax_table(bacteria_norm)[,"Species"]))
```

```
## [1] 1081
```

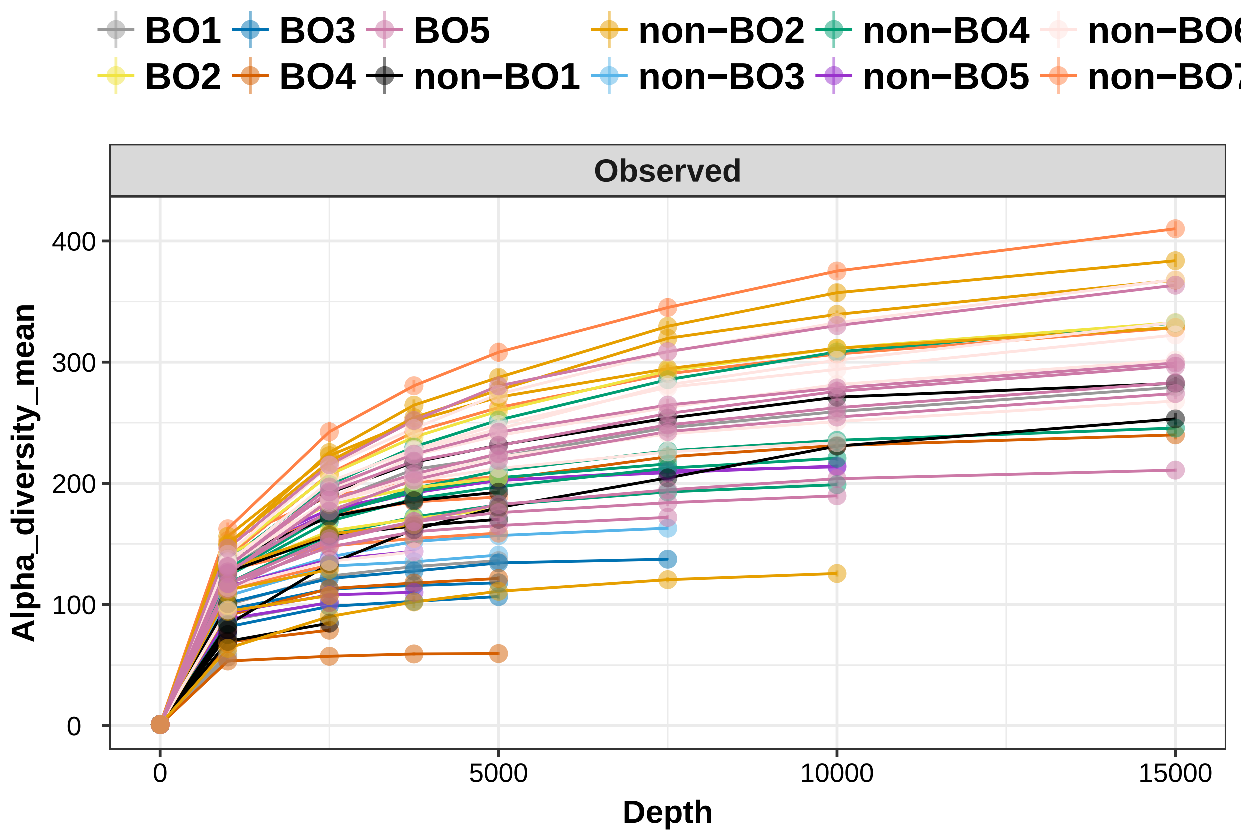
```r
#############################################################
######## plot number of microbial reads per sample ##########
#############################################################
df = as.data.frame(apply(otu_table(microbial), 2, sum))
colnames(df)[1]="count"
write.csv(df, file="Number_microbial_reads_per_sample.csv")
df$SampleIDFP = rownames(df)

df_merge = merge(df, data.frame(sample_data(microbial)), by.x = 'SampleIDFP',
                 by.y = 'row.names')
df_merge$id_order = paste(df_merge$AnnotatedID, df_merge$TimePoint, sep = ".")
df_selected = df_merge[,c("id", "id_order", "AnnotatedID", "count", "Outcome",
                          "SUBJECT.WUTX")]
h = median(df_selected$count)

#jpeg("LgTxCF12_microbial_readCounts.jpg", res = 300, height = 15, width = 30, units = 'cm')
ggplot(df_selected, aes(x = id_order, y = count, fill = AnnotatedID)) +
  geom_bar(stat="identity", position="stack") +
  ggtitle("# microbial reads per sample") +
  scale_fill_manual(values = cbbPalette_12) +
  labs(y = "# reads", x = "Sample") + theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=90, hjust=1,
                                  vjust=0.5, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=1,
                                  vjust=1, face="bold"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                   vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                   vjust=.5, face="bold"),
        legend.text=element_text(size=15, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position = "bottom",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(nrow = 2)) +
  geom_hline(yintercept=median(df_selected$count)) +
  geom_text(aes(0, h, label = h, vjust = 0, hjust = 0))
```



```
#dev.off()
```

```
#################################
###### Rarefaction Curves ######
#################################
calculate_rarefaction_curves <- function(psdata, measures, depths) {
  estimate_rarified_richness <- function(psdata, measures, depth) {
    if(max(sample_sums(psdata)) < depth) return()
    psdata <- prune_samples(sample_sums(psdata) >= depth, psdata)

    rarified_psdata <- rarefy_even_depth(psdata, depth, verbose = FALSE)
    alpha_diversity <- estimate_richness(rarified_psdata, measures = measures)

    # as.matrix forces the use of melt.array, which includes the Sample names (rownames)
    molten_alpha_diversity <- melt(as.matrix(alpha_diversity),
                                   varnames = c('Sample', 'Measure'), value.name = 'Alpha_diversity')
    molten_alpha_diversity
  }

  names(depths) <- depths # this enables automatic addition of the Depth to the output by ldply
  rarefaction_curve_data <- ldply(depths, estimate_rarified_richness, psdata = psdata,
                                  measures = measures, .id = 'Depth',
                                  .progress = ifelse(interactive(), 'text', 'none'))

  # convert Depth from factor to numeric
  rarefaction_curve_data$Depth <- as.numeric(levels(rarefaction_curve_data$Depth))[rarefaction_curve_dat
  rarefaction_curve_data
}
```

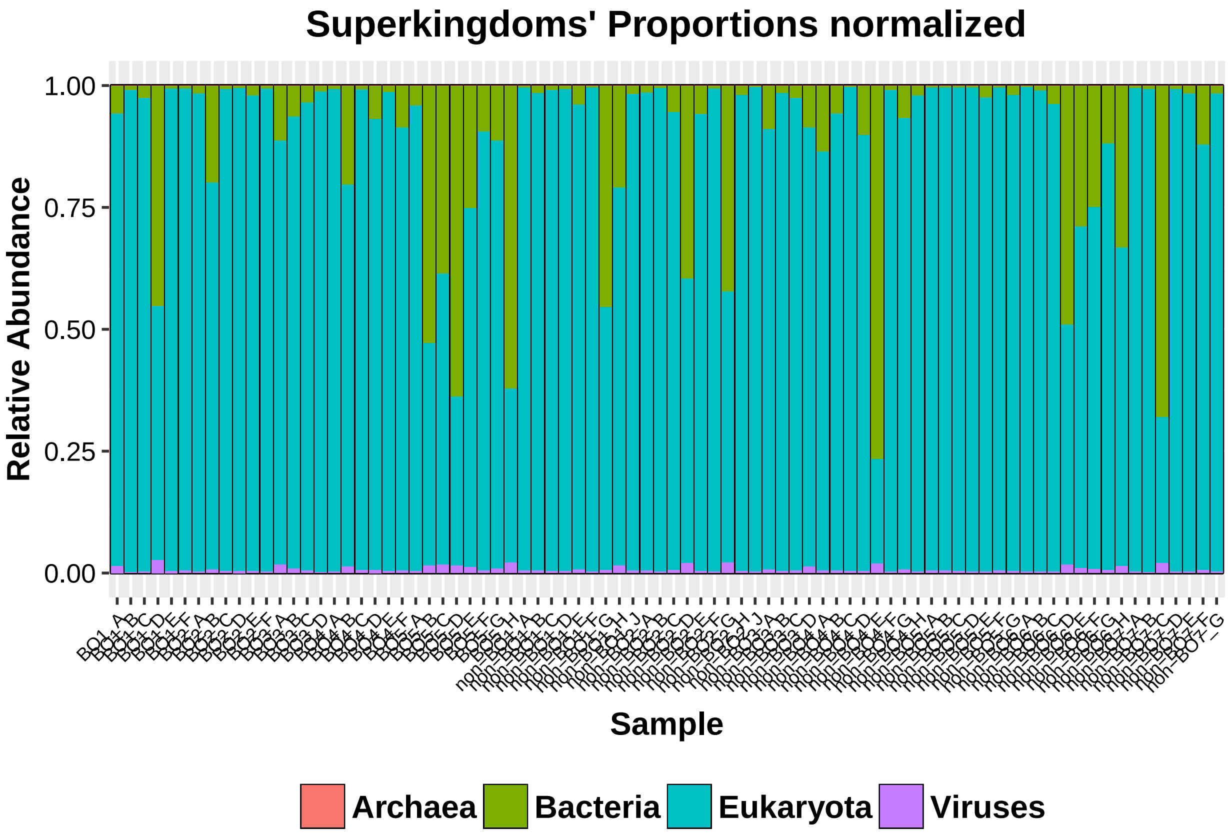
```
## Generate Rarefied Data
rarefaction_curve_data = calculate_rarefaction_curves(microbial, c('Observed'),
                                                      rep(c(1, 1000, 2500, 3750, 5000, 7500, 10000, 150
#summary(rarefaction_curve_data)
#max(sample_sums(microbial))
rarefaction_curve_data_summary = ddply(rarefaction_curve_data, c('Depth', 'Sample', 'Measure'),
                                        summarise, Alpha_diversity_mean = mean(Alpha_diversity),
                                        Alpha_diversity_sd = sd(Alpha_diversity))
rarefaction_curve_data_summary_verbose = merge(rarefaction_curve_data_summary,
                                               data.frame(sample_data(microbial)),
                                               by.x = 'Sample', by.y = 'row.names')

#jpeg("LgTxCF12_RarefactionCurve_0.4.jpg", res = 300, height = 15, width = 20, units = 'cm')
ggplot( data = rarefaction_curve_data_summary_verbose,
             mapping = aes(x = Depth, y = Alpha_diversity_mean,
                ymin = Alpha_diversity_mean - Alpha_diversity_sd,
                ymax = Alpha_diversity_mean + Alpha_diversity_sd, colour = AnnotatedID,
                group = Sample)) +
  geom_line() + geom_pointrange(size = 0.5, alpha = 0.5) +
  scale_color_manual(values = cbbPalette_12) +
  facet_wrap(facets = ~ Measure,  scales = 'free_y') +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="plain"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="top",
        strip.text = element_text(size=12, face = "bold"),
        legend.text = element_text(size=14, face="bold"), legend.title = element_blank()) +
        guides(colour = guide_legend(nrow = 2))
```



```
#dev.off()
```

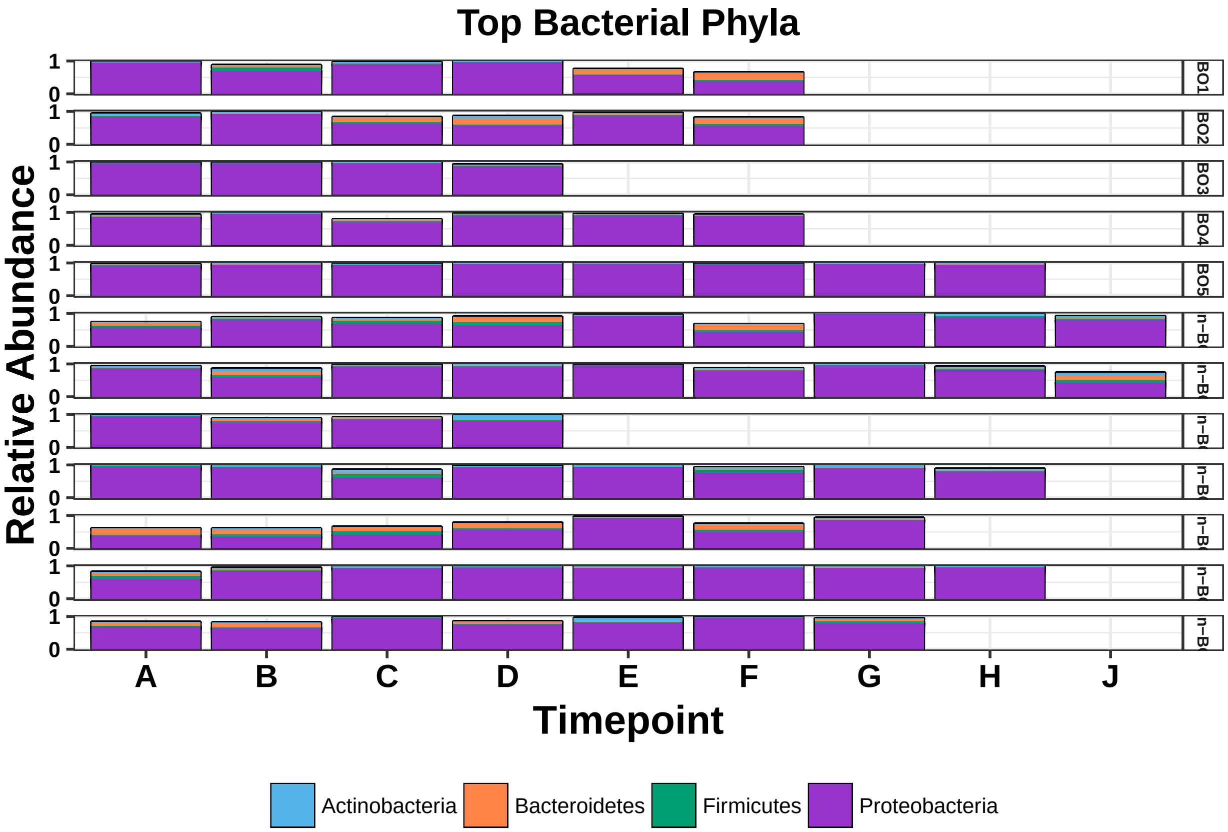
Taxonomic summaries:

```
#################################################
### Visualize superkingdoms proprtions   ####
#################################################
physeqRA = transform_sample_counts(physeq_norm, function(x) x / sum(x))
physeqRA.glom = tax_glom(physeqRA, "Superkingdom")
#jpeg("LgTxCF12_all_superkingdom_RA.jpg", res = 300, height = 15, width = 30, units = 'cm')
p = plot_bar(physeqRA.glom, x = "AnnotatedID_Timepoint", fill = "Superkingdom")#, facet_grid = ~Outcome
p + geom_bar(stat="identity", position="stack") + theme(legend.position="bottom") +
  ggtitle("Superkingdoms' Proportions normalized") +
  labs(y = "Relative Abundance", x = "Sample") +
  theme(axis.text.x = element_text(colour="black", size=7, angle=45, hjust=1,
                                   vjust=1, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="plain"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=12, face="bold"), legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```



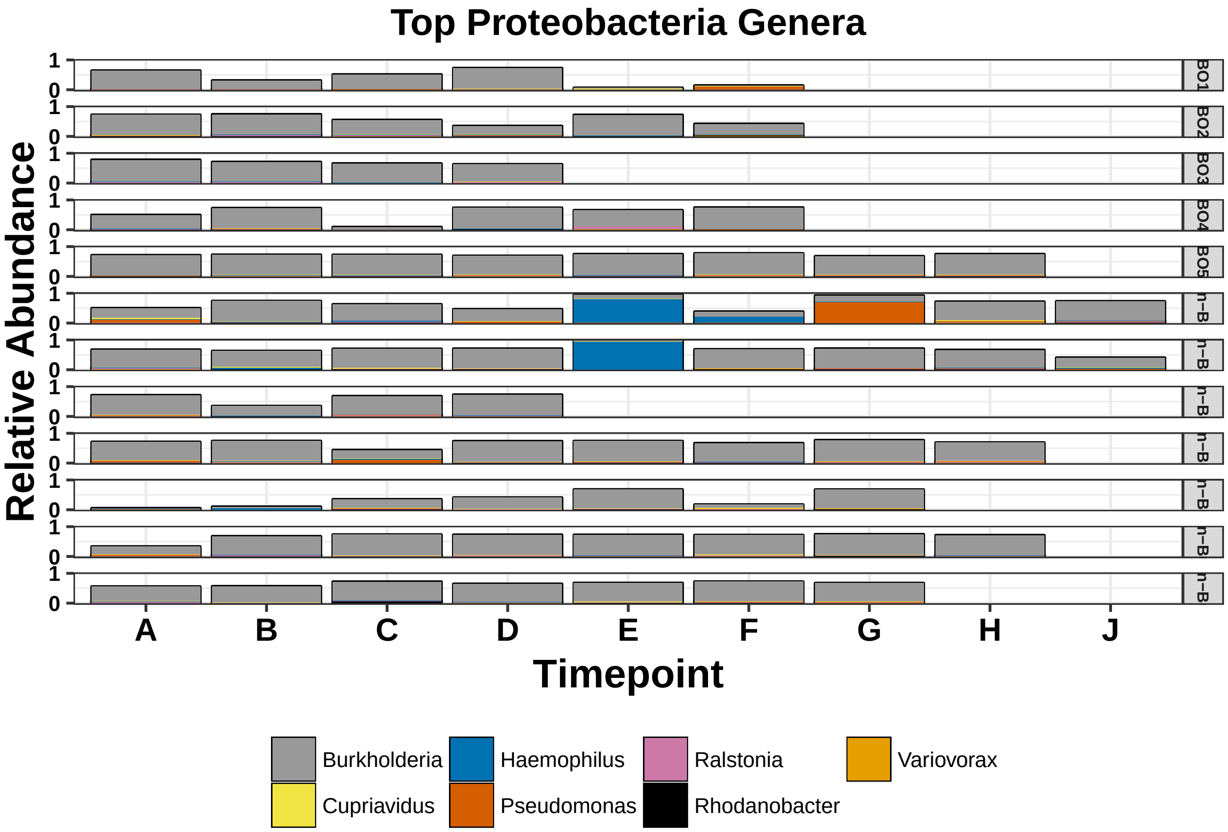
```
#dev.off()
```

```
######################################
### Subset top bacterial phyla   ####
######################################
top5ph = sort(tapply(taxa_sums(bacteria_norm), tax_table(bacteria_norm)[, "Phylum"], sum),
              decreasing = TRUE)[1:5]

top4ph = top5ph[which(names(top5ph)!="")] ### remove unannotated phylum
bacteria.RA = transform_sample_counts(bacteria_norm, function(x) x / sum(x))
bacteria.top4.phylum.RA = subset_taxa(bacteria.RA, Phylum %in% names(top4ph))

#jpeg("LUTX_top_4_bacterial_phyla.jpg", res = 300, height = 20, width = 12, units = 'cm')
p = plot_bar(bacteria.top4.phylum.RA, x = "TimePoint", fill="Phylum",
             facet_grid= AnnotatedID~.)
p + geom_bar(stat="identity", position="stack") +
  ggtitle("Top Bacterial Phyla") +
  scale_fill_manual(values = cbbPalette_9) +
  labs(y = "Relative Abundance", x = "Timepoint") + theme_bw() +
  scale_y_continuous(breaks = c(0, 1)) +
  theme(axis.text.x = element_text(colour="black", size=12, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=8, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=8, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="bottom",
        strip.text = element_text(size=6, face = "bold"), strip.background =element_rect(aes(fill=bacte
```



```
#dev.off()
```

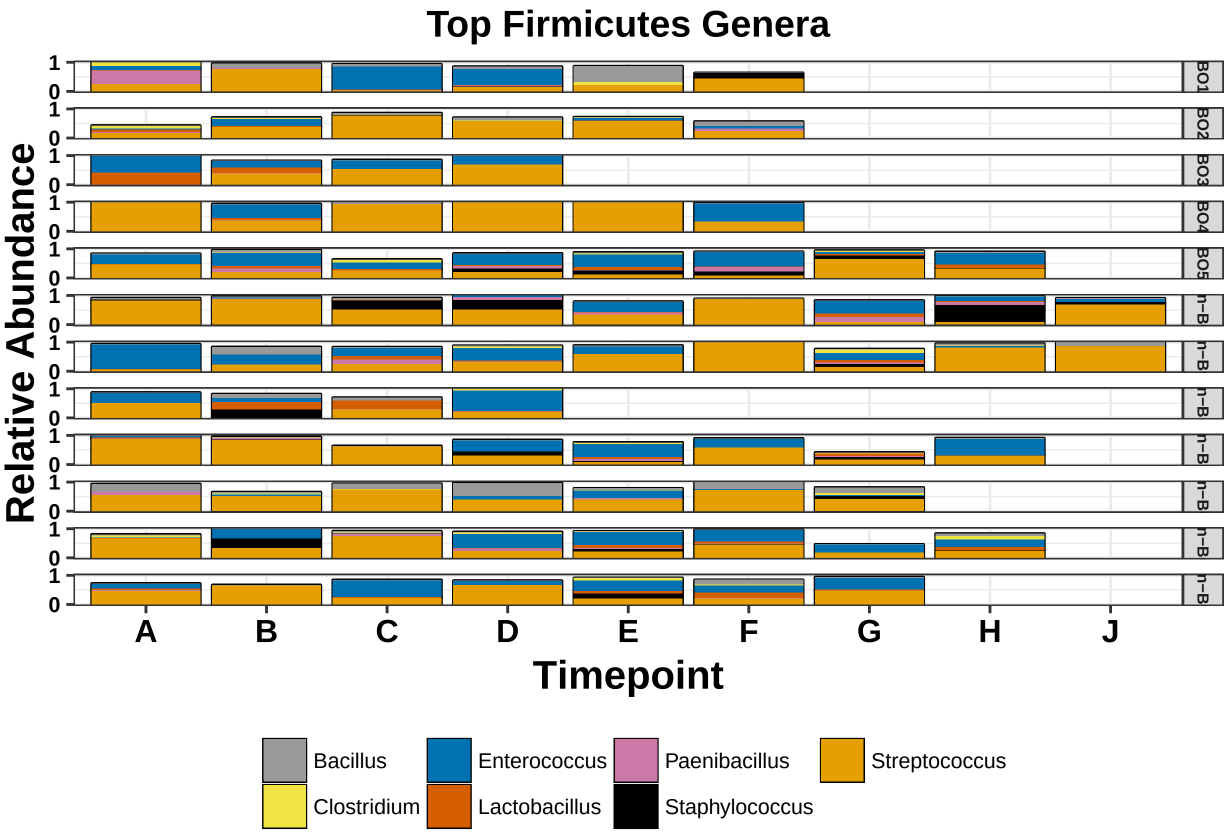
```
##############################################
#### Subset top Proteobacteria genera  #####
##############################################
proteo = subset_taxa(bacteria_norm, Phylum == "Proteobacteria")
proteo_genus = tax_glom(proteo, taxrank="Genus")
top9genus = sort(tapply(taxa_sums(proteo_genus), tax_table(proteo_genus)[, "Genus"], sum),
                 decreasing = TRUE)[1:9]
## remove Paraburkholderia as it is very homologous to Burkholderia,
# and Alteromonas as it's very likely to be contamination
top7genus = top9genus[-c(2,5)]
proteo.genus.RA = transform_sample_counts(proteo_genus, function(x) x / sum(x))
proteo.genus.RA.top8 = subset_taxa(proteo.genus.RA, Genus %in% names(top7genus))

#jpeg("LgTxCF12_top_8_proteo_genera.jpg", res = 300, height = 20, width = 12, units = 'cm')
p = plot_bar(proteo.genus.RA.top8, x = "TimePoint", fill="Genus",
             facet_grid= AnnotatedID ~ .)
p + geom_bar(stat="identity", position="stack") +
  ggtitle("Top Proteobacteria Genera") +
  scale_fill_manual(values = cbbPalette_10) +
  labs(y = "Relative Abundance", x = "Timepoint") + theme_bw() +
  scale_y_continuous(breaks = c(0, 1)) +
  theme(axis.text.x = element_text(colour="black", size=12, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=8, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=8, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="bottom",
        strip.text = element_text(size=6, face = "bold"))
```



```
#dev.off()
```

```
#####################################
### Subset top Firmicutes genera ###
#####################################
firm = subset_taxa(bacteria_norm, Phylum == "Firmicutes")
firm_genus = tax_glom(firm, taxrank="Genus")
top7genus = sort(tapply(taxa_sums(firm_genus), tax_table(firm_genus)[, "Genus"], sum),
                 decreasing = TRUE)[1:7]
firm.genus.RA = transform_sample_counts(firm_genus, function(x) x / sum(x))
firm.genus.RA.top8 = subset_taxa(firm.genus.RA, Genus %in% names(top7genus))
```

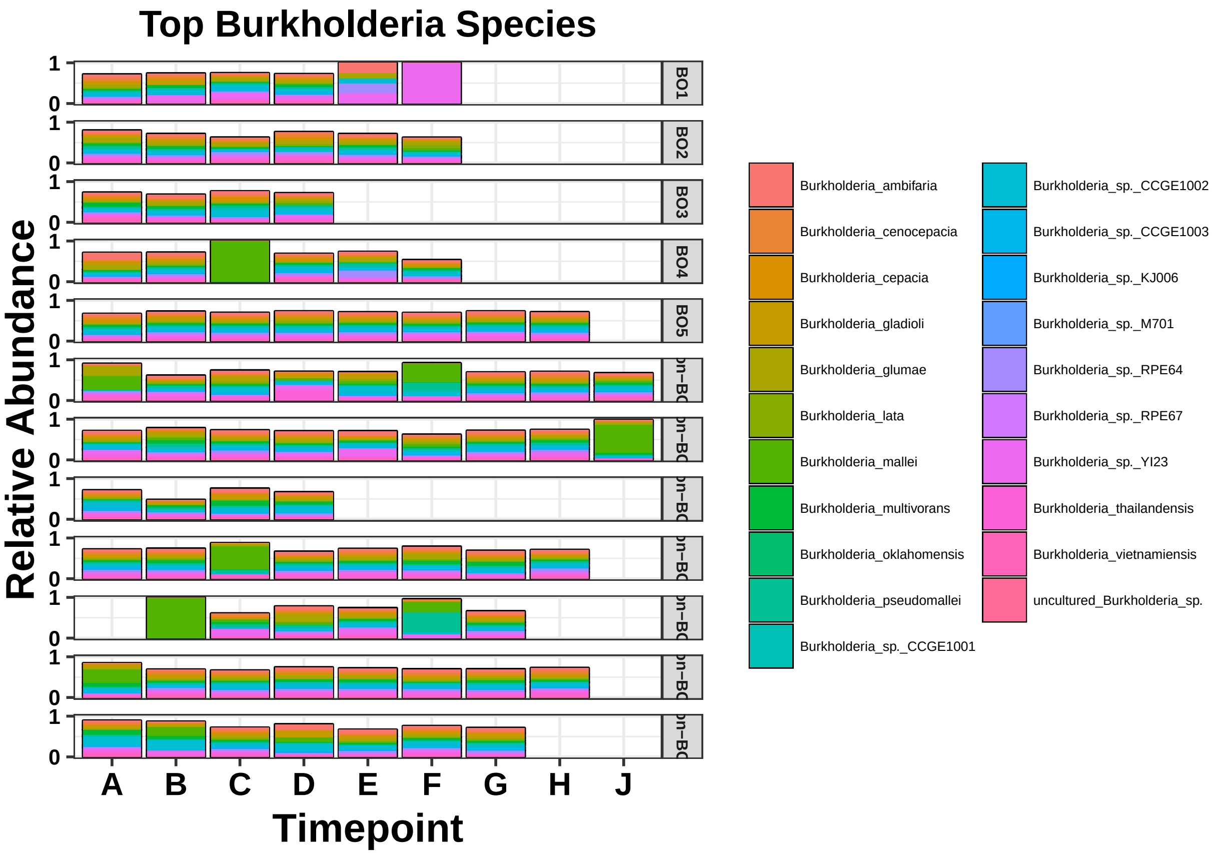
```
#jpeg("LgTxCF12_top_8_firm_genera.jpg", res = 300, height = 20, width = 12, units = 'cm')
p = plot_bar(firm.genus.RA.top8, x = "TimePoint", fill="Genus",
             facet_grid= AnnotatedID ~ .)
p + geom_bar(stat="identity", position="stack") +
  ggtitle("Top Firmicutes Genera") +
  scale_fill_manual(values = cbbPalette_10) +
  labs(y = "Relative Abundance", x = "Timepoint") + theme_bw() +
  scale_y_continuous(breaks = c(0, 1)) +
  theme(axis.text.x = element_text(colour="black", size=12, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=8, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=8, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="bottom",
        strip.text = element_text(size=6, face = "bold"))
```



```
#dev.off()
```

```
#######################################################
### Subsettop 10 Burkholderia species Normalized  ###
#######################################################
```

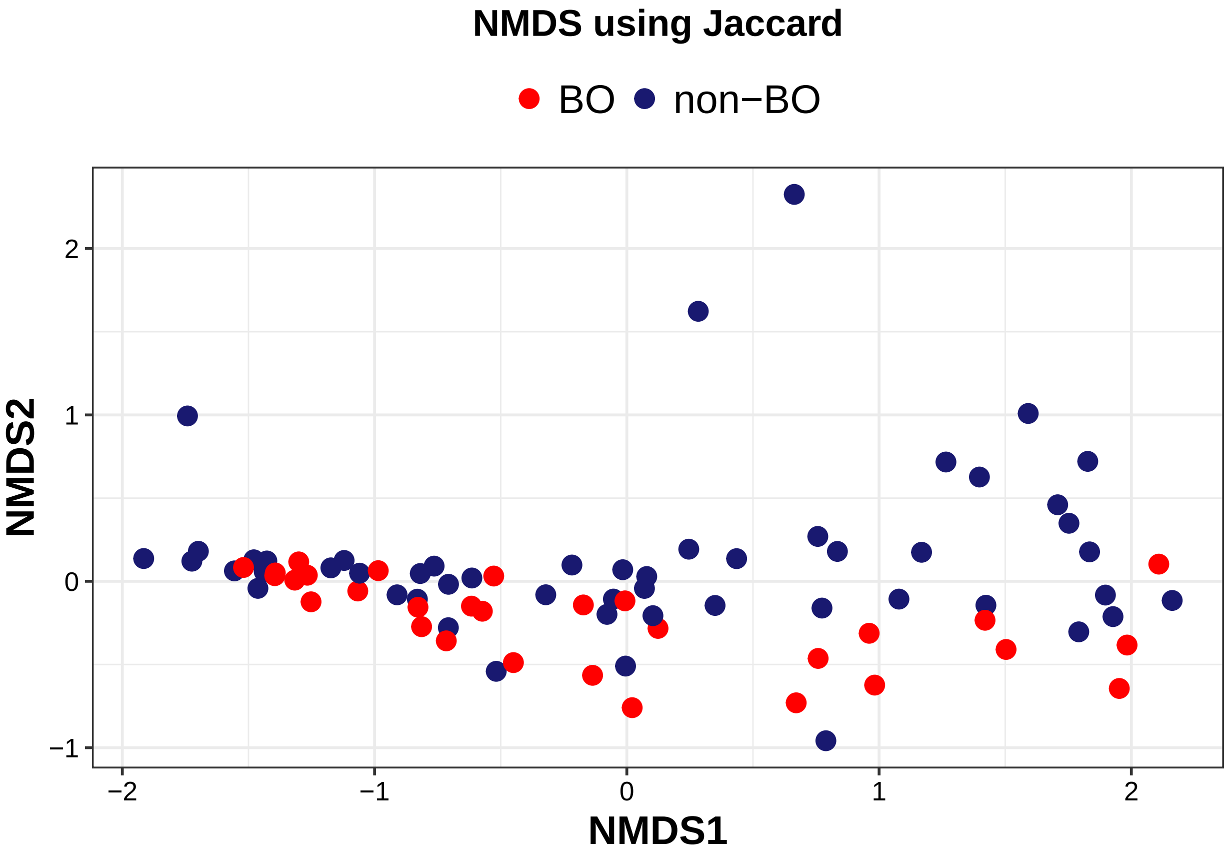
```
burkholderia = subset_taxa(bacteria_norm, Genus == "Burkholderia")
burkholderia.species = sort(tapply(taxa_sums(burkholderia), tax_table(burkholderia)[, "Species"], sum),
                            decreasing = TRUE)
burkholderia.species = burkholderia.species[which(names(burkholderia.species)!="")]
burkholderia.species.RA = transform_sample_counts(burkholderia, function(x) x / sum(x))
burkholderia.species.top = subset_taxa(burkholderia.species.RA, Species %in% names(burkholderia.species

## Save Burk Species
write.csv(otu_table(burkholderia.species.top), file = "burkholderia.species.top_OTUs.csv")
write.csv(tax_table(burkholderia.species.top), file="burkholderia.species.top_taxa.csv")
save(burkholderia.species.top, file = "proteo.genus.RA.top8.RData")


#jpeg("LgTxCF12_Burk_species.jpg", res = 300, height = 20, width = 20, units = 'cm')
p = plot_bar(burkholderia.species.top, x = "TimePoint", fill="Species",
             facet_grid= AnnotatedID ~ .)
p + geom_bar(stat="identity", position="stack") +
  ggtitle("Top Burkholderia Species") +
  #scale_fill_manual(values = cbbPalette_10) +
  labs(y = "Relative Abundance", x = "Timepoint") + theme_bw() +
  scale_y_continuous(breaks = c(0, 1)) +
  theme(axis.text.x = element_text(colour="black", size=12, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=8, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=5, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="right",
        strip.text = element_text(size=6, face = "bold"))
```



```
#dev.off()
```

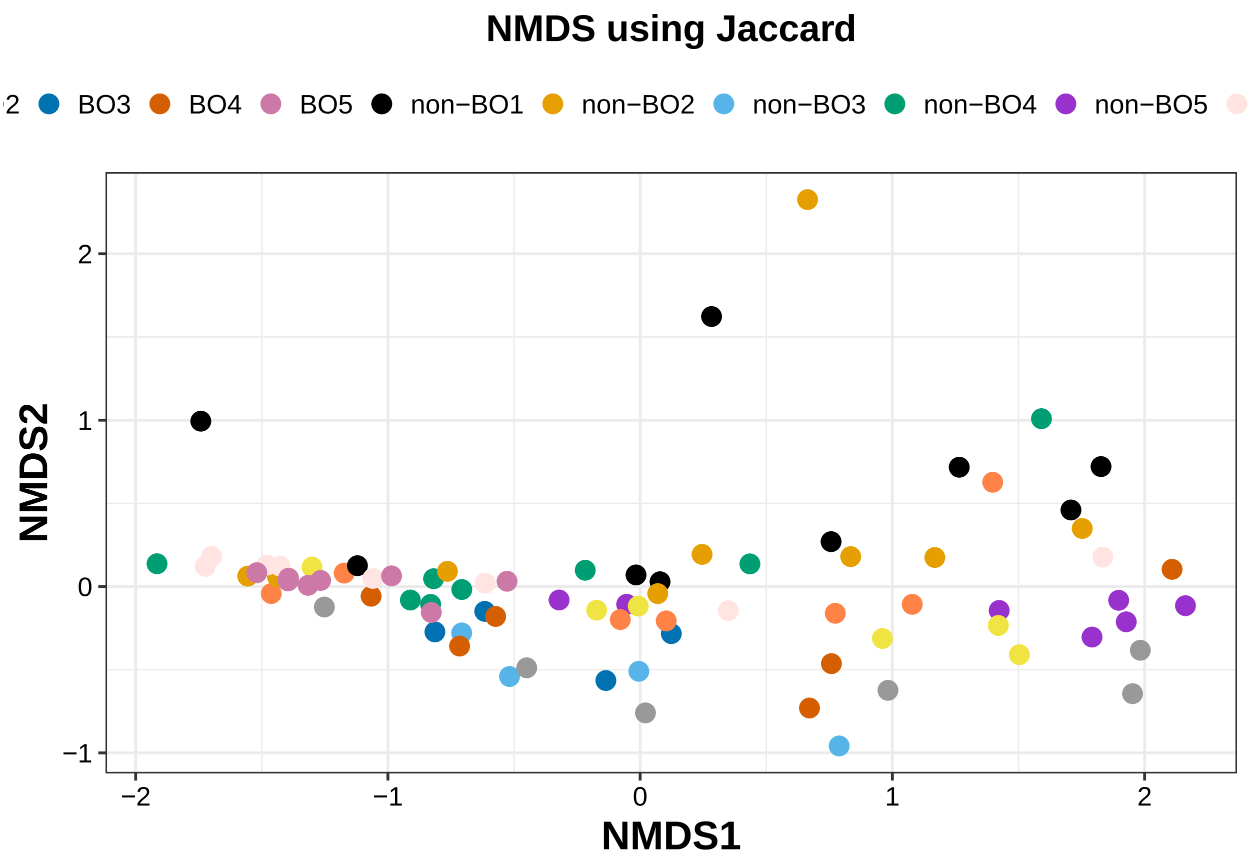
Ordination:

```
## NMDS (non-parametric Multi-Dimentional Scaling)
ord_nmds_jaccard = ordinate(microbial_norm, method = "NMDS", distance = "jaccard",
                            autotransform = FALSE, diag = TRUE, upper = TRUE)

## Plot NMDS labled by group
#jpeg("LgTxCF12_ordination_nmds_jaccard_groups_v0.4.jpg", res = 300, height = 15, width = 15, units = '
plot_ordination(microbial_norm, ord_nmds_jaccard, color = "Outcome") +  geom_point(size = 3) +
  ggtitle("NMDS using Jaccard") +
  scale_fill_manual(values = cbbPalette_12) +
  scale_color_manual(values=c( "red", "midnightblue"),  breaks=c("BO", "non-BO")) +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=1,
                                   vjust=0.5, face="plain"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=15, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position = "top",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(nrow = 1))
```



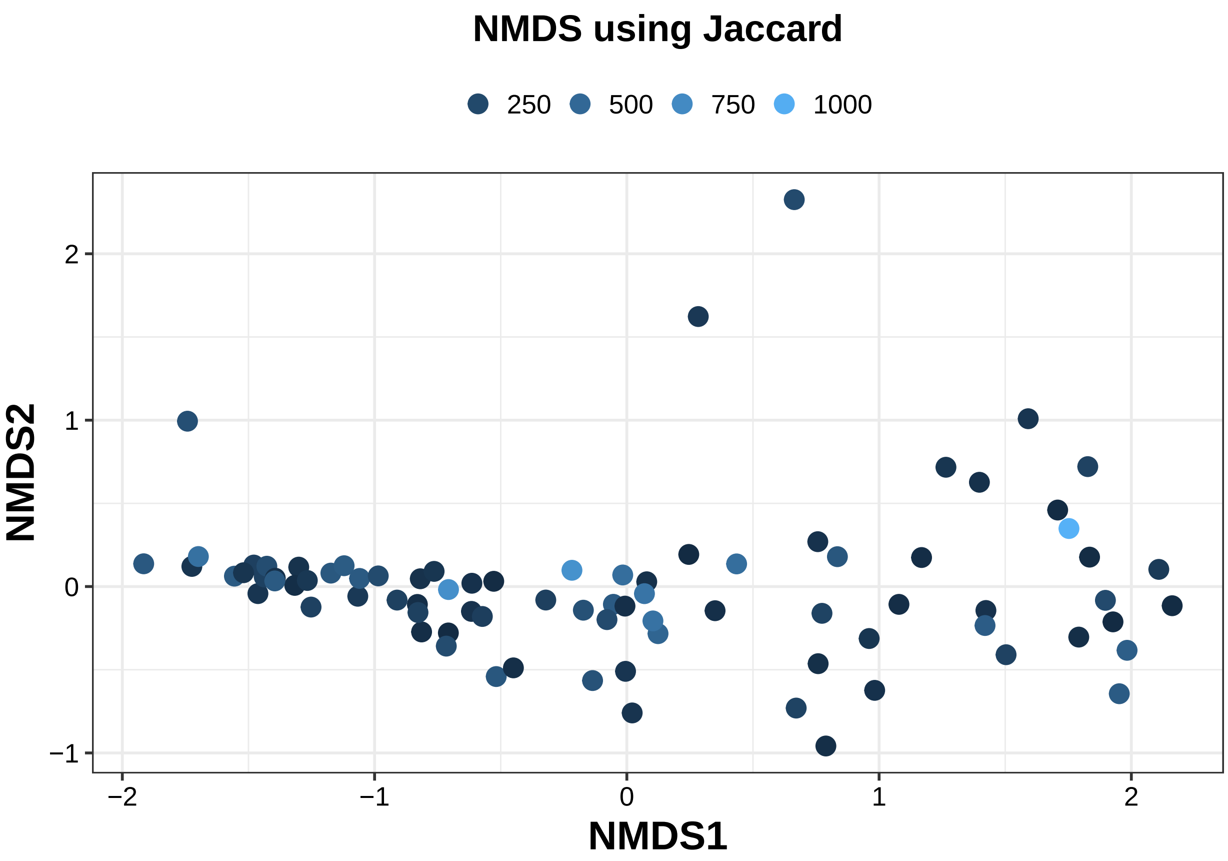
```
#dev.off()
```

```
## Plot NMDS labled by subject
#jpeg("LgTxCF12_ordination_nmds_jaccard_subjects_v0.4.jpg", res = 300, height = 15, width = 20, units =
plot_ordination(microbial_norm, ord_nmds_jaccard, color = "AnnotatedID") +
  geom_point(size = 3) +
  ggtitle("NMDS using Jaccard") +
  scale_fill_manual(values = cbbPalette_12) +
  scale_color_manual(values=cbbPalette_12) +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=1,
                                   vjust=0.5, face="plain"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position = "top",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(nrow = 1))
```



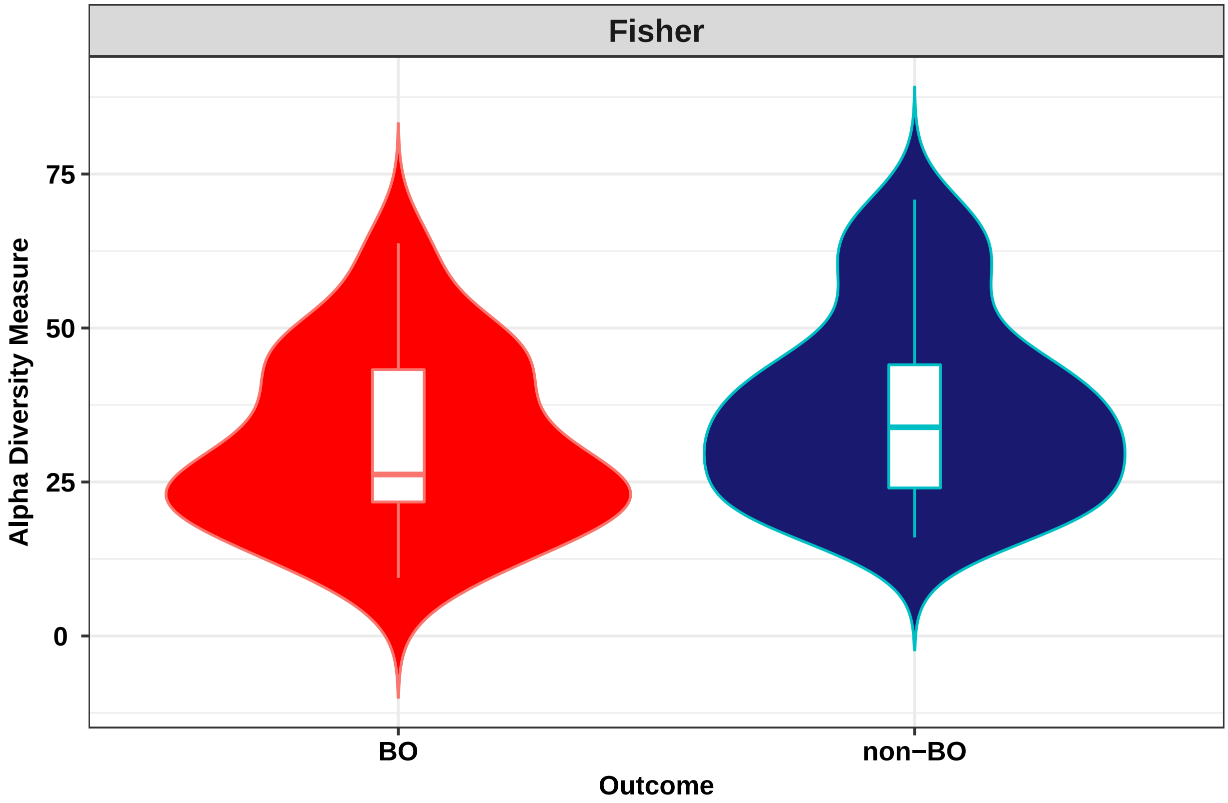
```
#dev.off()
```

```
## Plot NMDS labled by time
#jpeg("LgTxCF12_ordination_nmds_jaccard_subjects_v0.4.jpg", res = 300, height = 15, width = 20, units =
plot_ordination(microbial_norm, ord_nmds_jaccard, color = "TimeAfterTransplantation") +
  geom_point(size = 3) +
  ggtitle("NMDS using Jaccard") +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="plain"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=1,
                                   vjust=0.5, face="plain"),
        axis.title.x = element_text(colour="black", size=15, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=15, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position = "top",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(nrow = 1))
```



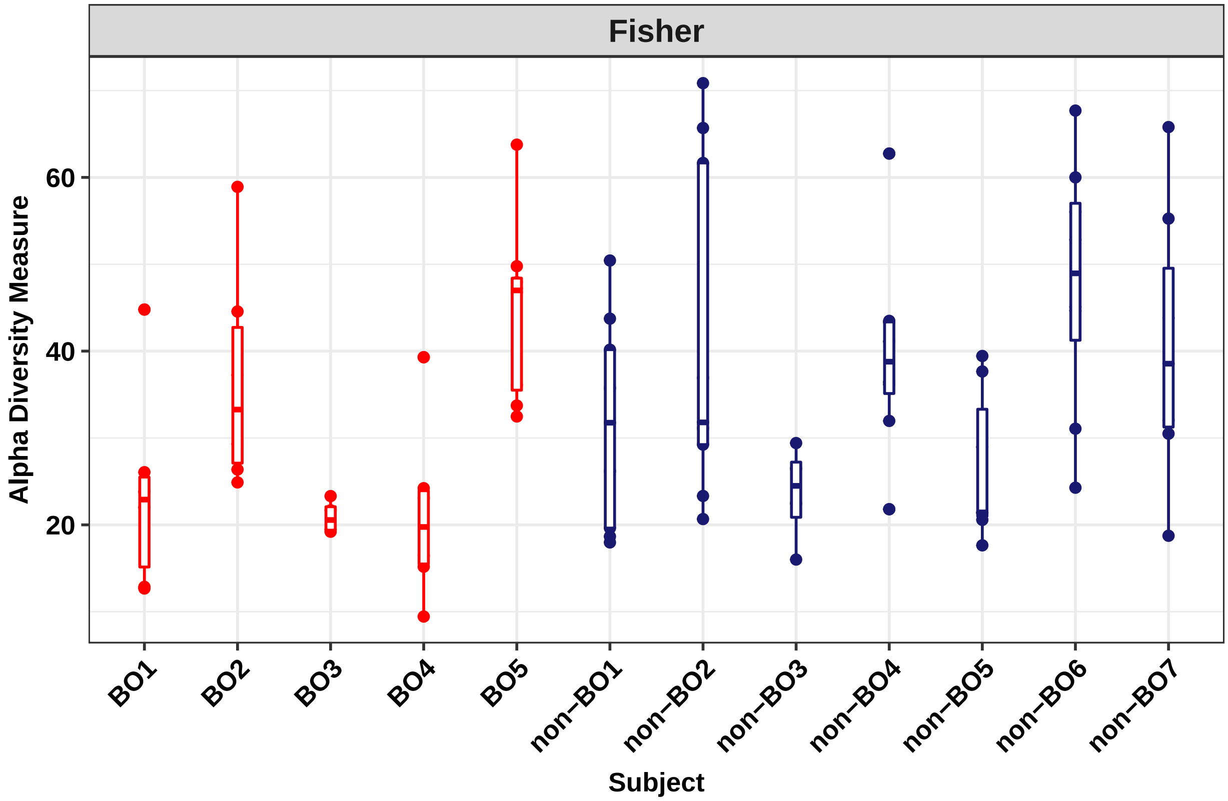
```
#dev.off()
```

```
#####################################################################################
##### Alpha Diversity using phyloseq using unnromalized/unfiltered count data ##########
#####################################################################################
#jpeg("LgTxCF12_alpha_diversity_violin_unfiltered_fisher.jpg", res = 300, height = 7, width = 5, units
plot_richness(microbial, x = "Outcome", color = "Outcome",
              measures = c("Fisher")) +
  geom_violin(aes(fill = Outcome), trim = FALSE) +
  ggtitle("")+#"Fisher Diversity Index per Group") +
  scale_fill_manual(values=c( "red", "midnightblue"),  breaks=c("bos", "non-bos")) +

  theme_bw() + stat_summary(fun.y=mean, geom="point", size=2, color="black") + geom_boxplot(width=0.1) +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=10, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=10, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=8, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="none",
        strip.text = element_text(size=12, face = "bold"))
```



```
#dev.off()
```

```
########################################
######### Visualize diversity per subject
########################################
#jpeg("LgTxCF12_alpha_diversity_perSubject_Fisher.jpg", res = 300, height = 10, width = 10, units = 'cm
plot_richness(microbial, x = "AnnotatedID", color = "Outcome",
              measures = c("Fisher")) +
  ggtitle("") +
  scale_fill_manual(values=c( "red", "midnightblue"),  breaks=c("bos", "non-bos")) +
  scale_color_manual(values=c( "red", "midnightblue"),  breaks=c("bos", "non-bos")) +
  labs( x = "Subject") +
  theme_bw() + stat_summary(fun.y=mean, geom="point", size=1, color="black") +
  geom_boxplot(width=0.1) +
  theme(axis.text.x = element_text(colour="black", size=10, angle=45, hjust=1,
                                   vjust=1, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=10, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=10, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=15, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        legend.position="none",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(nrow = 1))
```



```
#dev.off()
```

```
################################################################
######### Visualize diversity Tragectories per subject  #######
################################################################
### Plot for the first two timepoints and last timepoint
## merge meta and richness
richness = estimate_richness(microbial)
richness$SampleIDFP = rownames(richness)
meta_diversity = merge(meta,richness, by="SampleIDFP")
write.csv(meta_diversity, file="meta_diversity.csv", row.names = FALSE)


######################
## Diversity Stats
######################
myFun <- function(x) {
  median = median(x)
}

x = tapply(meta_diversity$Fisher_Diversity, meta_diversity$SUBJECT.WUTX, myFun)
x = as.data.frame(x)
x$SUBJECT.WUTX = rownames(x)
y = unique(meta_diversity[,c("SUBJECT.WUTX", "OutcomeMod")])
z = merge(x,y, by = "SUBJECT.WUTX")
wilcox.test(x~OutcomeMod, z)
```

```
##
##  Wilcoxon rank sum test
```

```
##
## data:  x by OutcomeMod
## W = 11, p-value = 0.3434
## alternative hypothesis: true location shift is not equal to 0
```

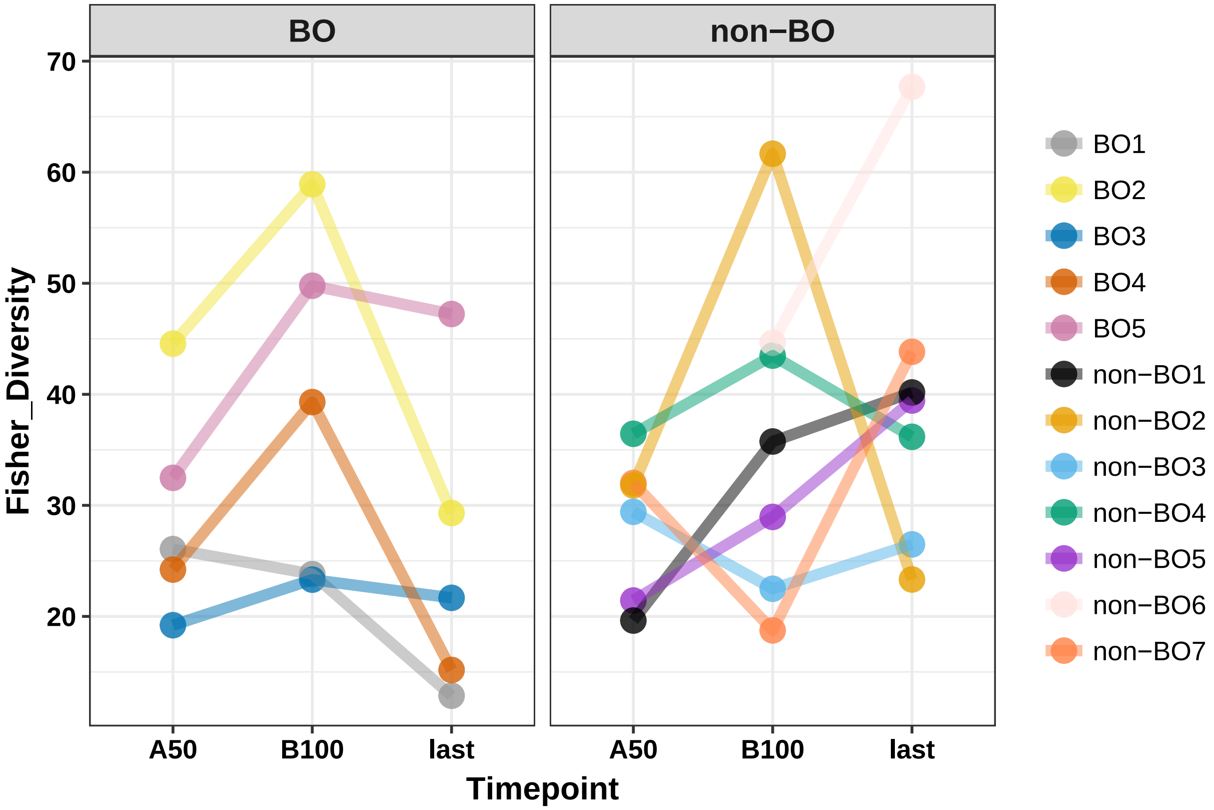
```r
median(z[z[,"OutcomeMod"]=="BO",]$x)
```

```
## [1] 22.90678
```

```r
median(z[z[,"OutcomeMod"]=="non-BO",]$x)
```

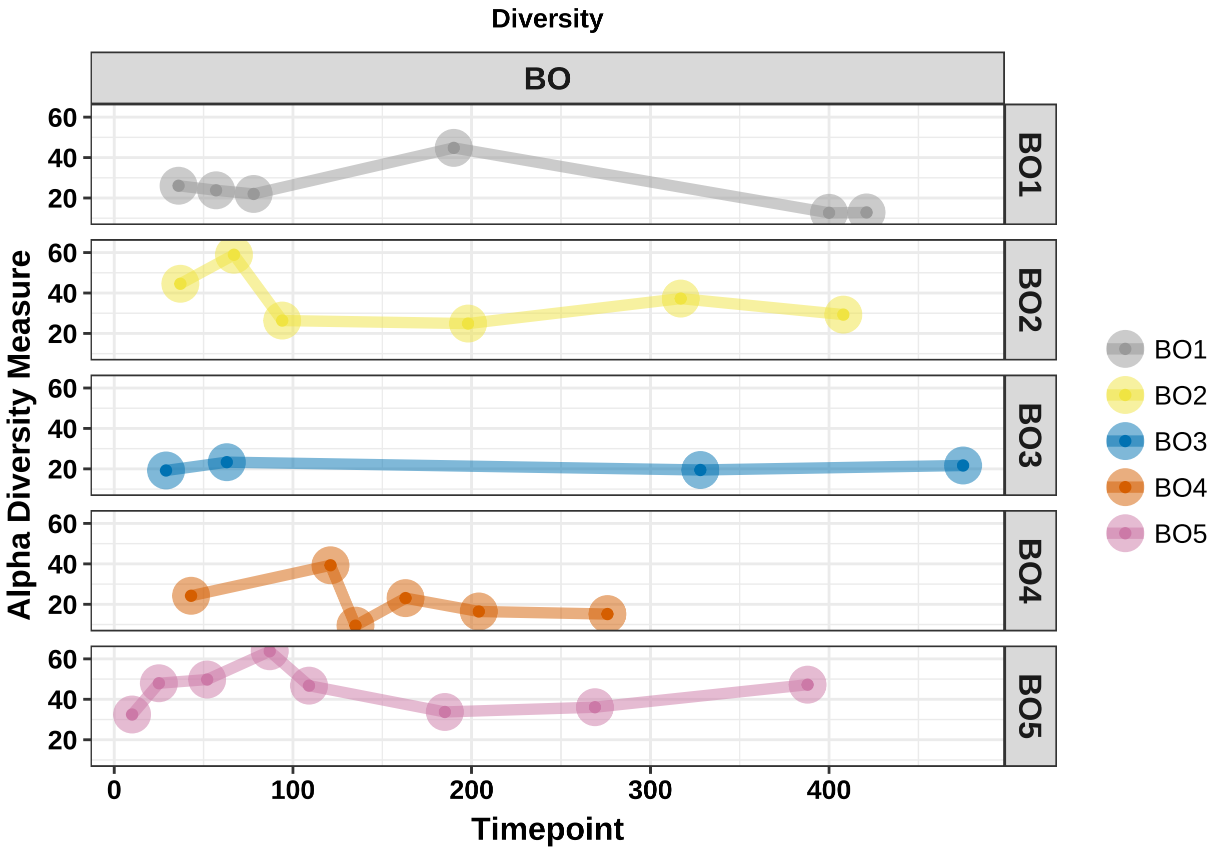
```
## [1] 31.79633
```

```r
################ Plot diversity for day50, day100, and last timepoint
sub3 = subset(meta, bin_status_mod == "A50" | bin_status_mod == "B100" | bin_status == "last")
#jpeg("LgTxCF12_alpha_diversity_unfiltered_BOS_3timepoints_bos_nonbos.jpg", res = 300, height = 10, wid
ggplot(sub3, aes(x = bin_status_mod, y=Fisher_Diversity, color = AnnotatedID,
                 group = AnnotatedID)) +
  geom_point(size=4, alpha=0.8) + geom_line(alpha=0.5, size=2) +
  ggtitle("") +
  scale_fill_manual(values = cbbPalette_12) +
  scale_color_manual(values=cbbPalette_12) +
  labs(x = "Timepoint") +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="right",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(. ~ OutcomeMod)
```



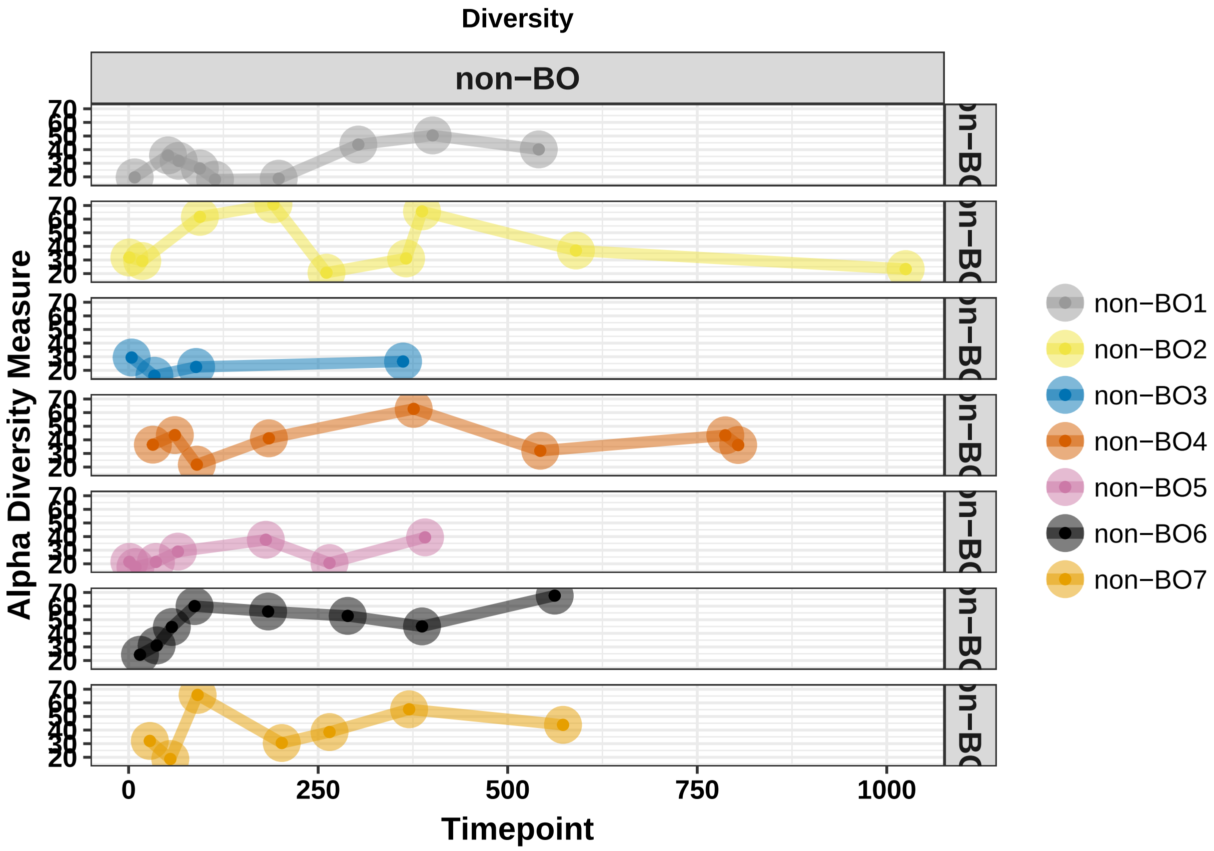
```
#dev.off()
```

```
########################################
## diversity unfiltered all time points
#########################################
# BO
sub2 = subset_samples(microbial, Outcome == "BO")
#jpeg("LgTxCF12_alpha_diversity_unfiltered_BOS_trajectory.jpg", res = 300, height = 15, width = 20, uni
plot_richness(sub2, x = "TimeAfterTransplantation", color = "AnnotatedID",
              measures = c("Fisher")) + geom_point(size=6, alpha=0.5) +
  geom_line(alpha=0.5, size=2, group = "AnnotatedID") +

  ggtitle("Diversity") +
  scale_fill_manual(values = cbbPalette_12) +
  scale_color_manual(values=cbbPalette_12) +

  labs(x = "Timepoint") +
  theme_bw() + #stat_summary(fun.y=mean, geom="point", size=1, color="black") +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="right",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(AnnotatedID ~ Outcome)
```



```
#dev.off()
```

```
## nonBO
sub2 = subset_samples(microbial, Outcome == "non-BO")
#jpeg("LgTxCF12_alpha_diversity_unfiltered_nonBOS_trajectory.jpg", res = 300, height = 20, width = 20,
plot_richness(sub2, x = "TimeAfterTransplantation", color = "AnnotatedID",
              measures = c("Fisher")) + geom_point(size=6, alpha=0.5) +
  geom_line(alpha=0.5, size=2, group = "AnnotatedID") +
  ggtitle("Diversity") +
  scale_fill_manual(values = cbbPalette_12) +
  scale_color_manual(values=cbbPalette_12) +
  labs(x = "Timepoint") +
  theme_bw() + #stat_summary(fun.y=mean, geom="point", size=1, color="black") +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="right",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(AnnotatedID ~ Outcome)
```



```
#dev.off()
```

```
############################################################
### Statistics on diversity as a group not longitudinal
############################################################
### TODO: Add pvalue to graphs
richness = estimate_richness(microbial)
mannwhitney_test = t(sapply(richness, function(x) unlist(wilcox.test(x ~ sample_data(microbial)$Outcome
t(sapply(richness, function(x) unlist(wilcox.test(x ~ sample_data(microbial)$Outcome))))
```

```
##              statistic.W p.value               null.value.location shift
## Observed     "653"       "0.223268188665942"   "0"
## Chao1        "652.5"     "0.221464114486526"   "0"
## se.chao1     "742.5"     "0.716232233634253"   "0"
## ACE          "653"       "0.223293379504406"   "0"
## se.ACE       "652"       "0.219658119573084"   "0"
## Shannon      "485"       "0.00458055445357641" "0"
## Simpson      "513"       "0.0103001835074059"  "0"
## InvSimpson   "513"       "0.0103001835074059"  "0"
## Fisher       "620"       "0.124660931880396"   "0"
##              alternative method
## Observed   "two.sided" "Wilcoxon rank sum test with continuity correction"
## Chao1      "two.sided" "Wilcoxon rank sum test with continuity correction"
## se.chao1   "two.sided" "Wilcoxon rank sum test with continuity correction"
## ACE        "two.sided" "Wilcoxon rank sum test with continuity correction"
## se.ACE     "two.sided" "Wilcoxon rank sum test with continuity correction"
## Shannon    "two.sided" "Wilcoxon rank sum test with continuity correction"
## Simpson    "two.sided" "Wilcoxon rank sum test with continuity correction"
## InvSimpson "two.sided" "Wilcoxon rank sum test with continuity correction"
## Fisher     "two.sided" "Wilcoxon rank sum test with continuity correction"
##            data.name
## Observed   "x by sample_data(microbial)$Outcome"
## Chao1      "x by sample_data(microbial)$Outcome"
## se.chao1   "x by sample_data(microbial)$Outcome"
## ACE        "x by sample_data(microbial)$Outcome"
## se.ACE     "x by sample_data(microbial)$Outcome"
## Shannon    "x by sample_data(microbial)$Outcome"
## Simpson    "x by sample_data(microbial)$Outcome"
## InvSimpson "x by sample_data(microbial)$Outcome"
## Fisher     "x by sample_data(microbial)$Outcome"
```

```
###########################################
##### Clinical association
###########################################
# Neutrophil_perc ~ Diversity
fit_bos <- lm(Neutrophil_perc ~ Fisher, subset = Outcome == "BO", data = meta_diversity)
summary(fit_bos)
```

Call: lm(formula = Neutrophil_perc ~ Fisher, data = meta_diversity, subset = Outcome == “BO”)

Residuals: Min 1Q Median 3Q Max -34.743 -13.404 -3.658 7.215 55.269

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 53.7486 11.9394 4.502 0.000116 * **Fisher -0.9895 0.3488 -2.837 0.008531** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ‘’ 1

Residual standard error: 26.69 on 27 degrees of freedom (1 observation deleted due to missingness) Multiple R-squared: 0.2297, Adjusted R-squared: 0.2011 F-statistic: 8.049 on 1 and 27 DF, p-value: 0.008531

```
fit_non_bos <- lm(Neutrophil_perc ~ Fisher, subset = Outcome == "non-BO",data = meta_diversity)
summary(fit_non_bos)
```

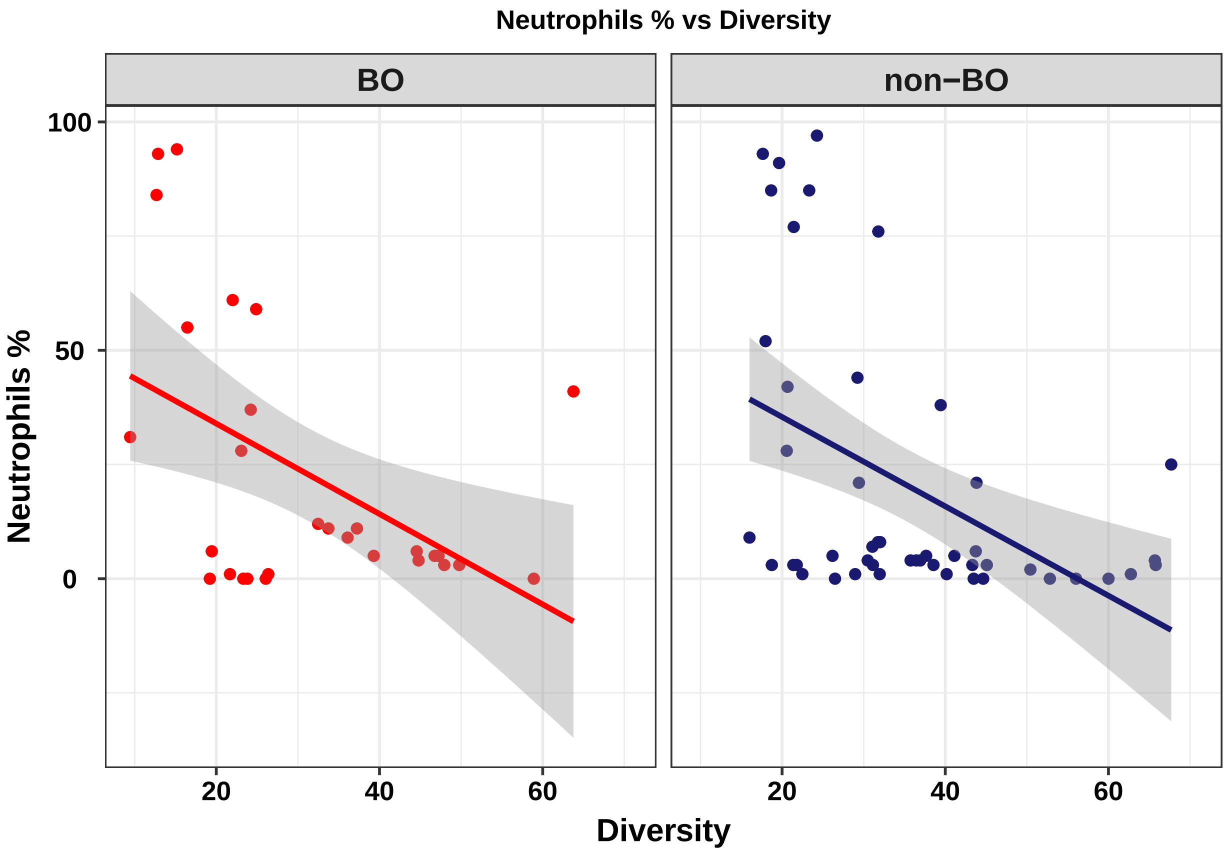
```
##
## Call:
## lm(formula = Neutrophil_perc ~ Fisher, data = meta_diversity,
##     subset = Outcome == "non-BO")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.629 -16.392  -8.729  12.606  65.779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.9595    10.7097   5.132 5.64e-06 ***
## Fisher       -0.9779     0.2818  -3.470  0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 27.28 on 46 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2075, Adjusted R-squared:  0.1903
## F-statistic: 12.04 on 1 and 46 DF,  p-value: 0.001141
```

```
#jpeg("LgTxCF12_Neutrophils__diversity_pergroup.jpg", res = 300, height = 8, width = 15, units = 'cm')
ggplot(meta_diversity, aes(x = Fisher, y = Neutrophil_perc, color = Outcome)) +
  geom_point() +
  stat_smooth(method = "lm", aes(color=Outcome)) +
  ggtitle("Neutrophils % vs Diversity") +
  scale_fill_manual(values=c( "red", "midnightblue"),  breaks=c("BO", "non-BO")) +
  scale_color_manual(values=c( "red", "midnightblue"),  breaks=c("BO", "non-BO")) +
  labs(x = "Diversity", y = "Neutrophils %") +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="none",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(. ~ Outcome)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



```
#dev.off()
```

```
#######  Diversity ~ Lymphocytes
fit_bos <- lm(Lymphocyte_perc ~ Fisher, subset = Outcome == "BO",data = meta_diversity)
summary(fit_bos)
```

```
##
## Call:
## lm(formula = Lymphocyte_perc ~ Fisher, data = meta_diversity,
##     subset = Outcome == "BO")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.387 -14.195  -8.253  10.725  62.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.7765     9.6123   1.433    0.163
## Fisher        0.2935     0.2820   1.041    0.307
##
## Residual standard error: 21.59 on 28 degrees of freedom
## Multiple R-squared:  0.03724,    Adjusted R-squared:  0.002853
## F-statistic: 1.083 on 1 and 28 DF,  p-value: 0.3069
```

```
fit_non_bos <- lm(Lymphocyte_perc ~ Fisher, subset = Outcome == "non-BO",data = meta_diversity)
summary(fit_non_bos)
```

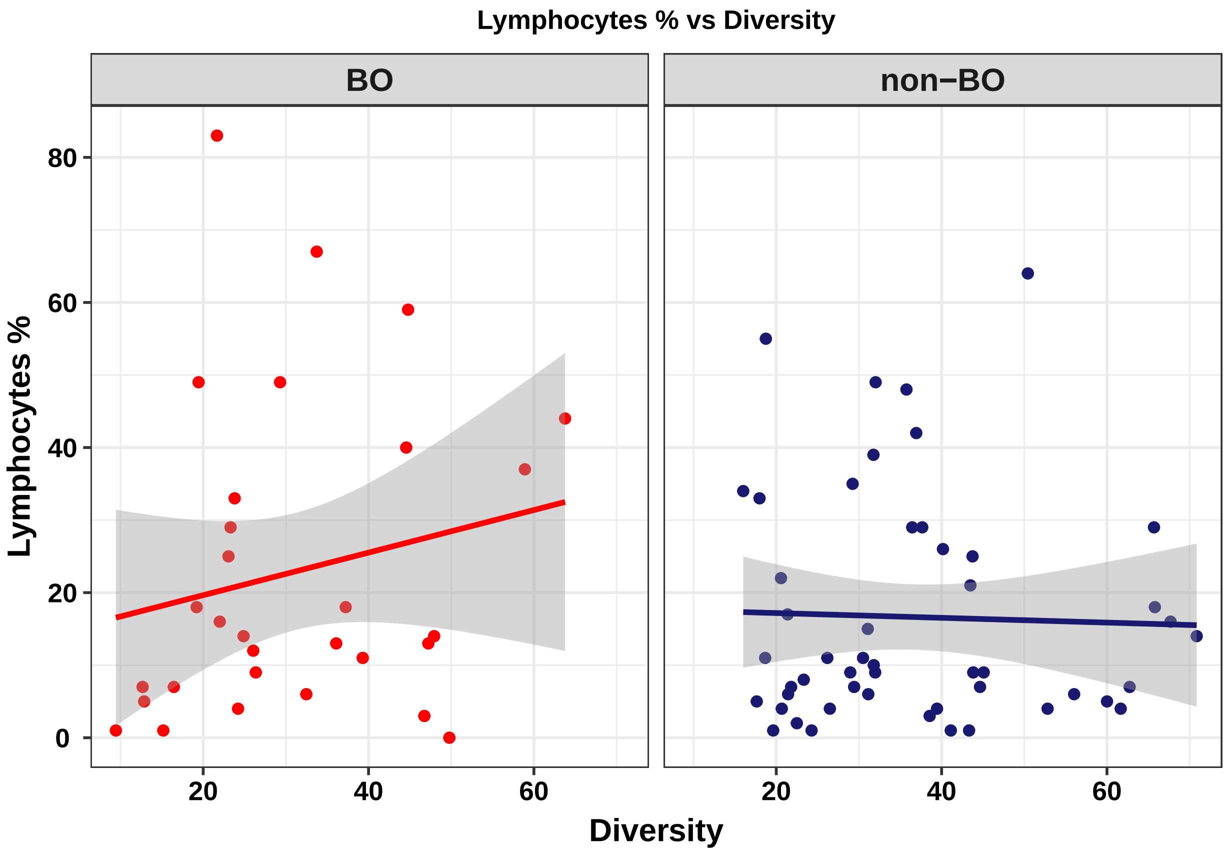
```
##
```

```
## Call:
## lm(formula = Lymphocyte_perc ~ Fisher, data = meta_diversity,
##     subset = Outcome == "non-BO")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.196 -11.071  -7.079   9.257  47.815
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.84057    5.90997   3.019  0.00406 **
## Fisher      -0.03282    0.14948  -0.220  0.82715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.85 on 48 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.001003,   Adjusted R-squared:  -0.01981
## F-statistic: 0.04821 on 1 and 48 DF,  p-value: 0.8271
```

```r
#jpeg("LgTxCF12_Lymphocytes__diversity_pergroup.jpg", res = 300, height = 8, width = 15, units = 'cm')
ggplot(meta_diversity, aes(x = Fisher, y = Lymphocyte_perc, color = Outcome)) +
  geom_point() +
  stat_smooth(method = "lm", aes(color=Outcome)) +
  ggtitle("Lymphocytes % vs Diversity") +
  scale_fill_manual(values=c( "red", "midnightblue"),  breaks=c("BO", "non-BO")) +
  scale_color_manual(values=c( "red", "midnightblue"),  breaks=c("BO", "non-BO")) +
  labs(x = "Diversity", y = "Lymphocytes %") +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="none",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(. ~ Outcome)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



```
#dev.off()
```

```
#######  Diversity ~ Nutrophils (categorized by microbiology status)
meta_diversity$culture ="+"
meta_diversity$culture[which(meta_diversity$LavageMicrobiology == "NoSignificantGrowth" | meta_diversity

fit_negCult <- lm(Neutrophil_perc ~ Fisher, subset = culture == "-", data = meta_diversity)
summary(fit_negCult)
```

```
##
## Call:
## lm(formula = Neutrophil_perc ~ Fisher, data = meta_diversity,
##     subset = culture == "-")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.722 -16.974  -5.759   1.279  73.752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.8943     9.9103   4.026 0.000206 ***
## Fisher       -0.6858     0.2536  -2.704 0.009499 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.43 on 47 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1347, Adjusted R-squared:  0.1162
## F-statistic: 7.314 on 1 and 47 DF,  p-value: 0.009499
```

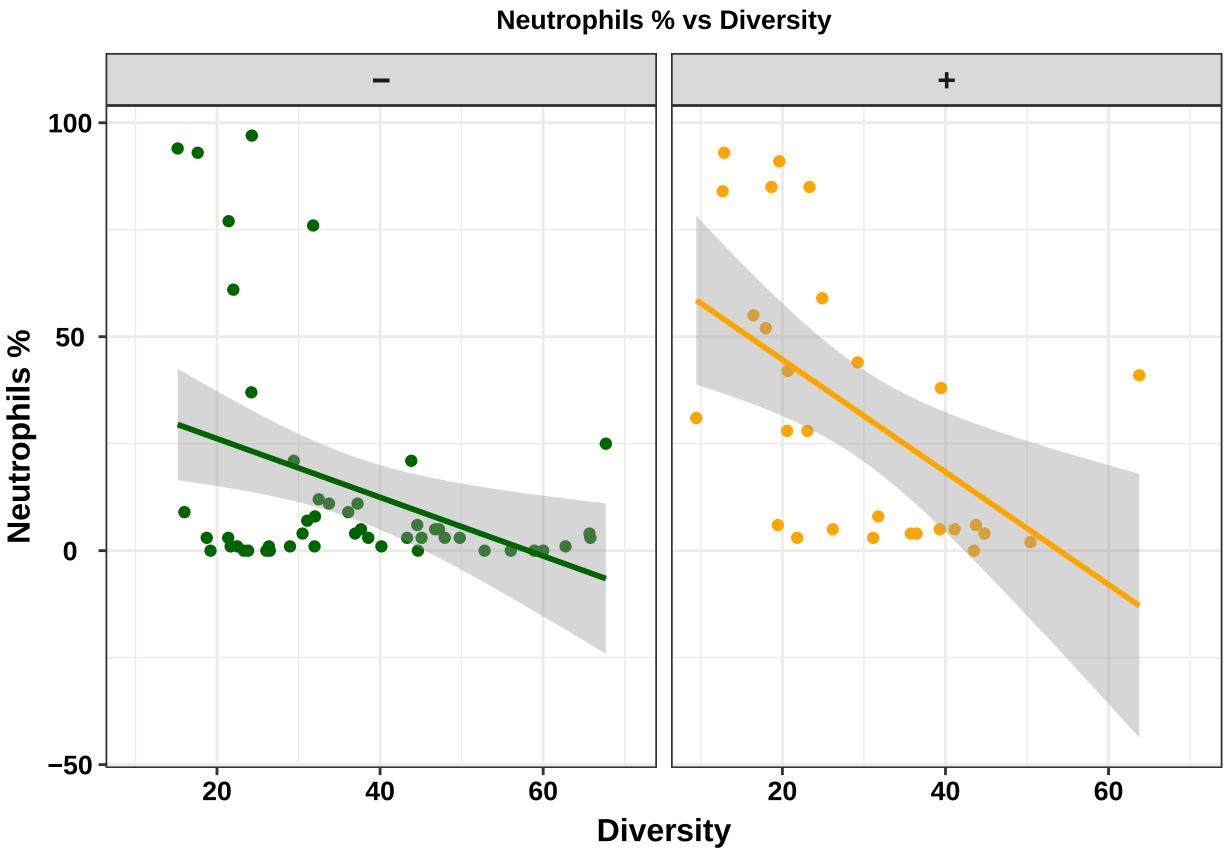
```
fit_posCult <- lm(Neutrophil_perc ~ Fisher, subset = culture == "+", data = meta_diversity)
summary(fit_posCult)
```

```
##
## Call:
## lm(formula = Neutrophil_perc ~ Fisher, data = meta_diversity,
##     subset = culture == "+")
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39.37 -19.26  -7.76  19.37  53.88
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.9296    12.9863   5.462 9.97e-06 ***
## Fisher       -1.3142     0.4072  -3.227  0.00337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.55 on 26 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.286,  Adjusted R-squared:  0.2585
## F-statistic: 10.41 on 1 and 26 DF,  p-value: 0.003367
```

```
#jpeg("LgTxCF12_Neutrophils__diversity_perCulture.jpg", res = 300, height = 8, width = 15, units = 'cm'
ggplot(meta_diversity, aes(x = Fisher, y = Neutrophil_perc, color = culture)) +
  geom_point() +
  stat_smooth(method = "lm", aes(color=culture)) +
  ggtitle("Neutrophils % vs Diversity") +
  scale_fill_manual(values=c( "darkgreen", "orange"),  breaks=c("+", "-")) +
  scale_color_manual(values=c("darkgreen", "orange"),  breaks=c("+", "-")) +
  labs(x = "Diversity", y = "Neutrophils %") +
  theme_bw() +
  theme(axis.text.x = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0, face="bold"),
        axis.text.y = element_text(colour="black", size=10, angle=0, hjust=0.5,
                                   vjust=0.5, face="bold"),
        axis.title.x = element_text(colour="black", size=12, angle=0, hjust=.5,
                                    vjust=0.5, face="bold"),
        axis.title.y = element_text(colour="black", size=12, angle=90, hjust=.5,
                                    vjust=.5, face="bold"),
        legend.text=element_text(size=10, face="plain"),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        legend.position="none",
        strip.text = element_text(size=12, face = "bold")) +
  guides(colour = guide_legend(ncol = 1)) + facet_grid(. ~ culture)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



```
#dev.off()
```

```
######################################################################################
######## MetaLonDA Analysis on Diversity/Clinical data #######
######################################################################################
## MetaLonDA on Diversity
count = meta_diversity$Fisher
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess",
                                points = points, text = "diversity", parall = FALSE,
                                pvalue.threshold = 0.1, adjust.method = "BH",
                                col = c("firebrick", "blue"), ylabel = "Diversity")
```

```
## MetaLonDA on FEV
count = meta_diversity$FEV1_Per_Predicted
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess",
                                points = points, text = "FEV1 Percent Predicted",
                                parall = FALSE, pvalue.threshold = 0.1, adjust.method = "BH",
                                col = c("firebrick", "blue"), ylabel = "FEV1 (%) Predicted")
```

```
## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 9 rows containing missing values (geom_path).

## Warning: Removed 17 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## MetaLonDA on FVC
count = meta_diversity$FVC_Per_Predicted
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess", points = points,
                                text = "FVC Percent Predicted", parall = FALSE,
                                pvalue.threshold = 0.1,  adjust.method = "BH",
                                col = c("firebrick", "blue"), ylabel = "FVC (%) Predicted")
```

```
## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 9 rows containing missing values (geom_path).

## Warning: Removed 17 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## MetaLonDA on FEV1/FVC
count = meta_diversity$FEV1_FVC_Ratio
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group, ID = id,
                                n.perm = 1000, fit.method = "lowess", points = points,
                                text = "FEV1 over FVC", parall = FALSE, pvalue.threshold = 0.1,
                                adjust.method = "BH", col = c("firebrick", "blue"),
                                ylabel = "FEV1/FVC")
```

```
## Warning: Removed 13 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing missing values (geom_path).

## Warning: Removed 19 rows containing missing values (geom_point).

## Warning: Removed 17 rows containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## MetaLonDA on LavageTotalCellCountcells_per_mcL
count = meta_diversity$LavageTotalCellCountcells_per_mcL
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
```

```
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess", points = points,
                                text = "Lavage Total CellCount cells_per_mcL", parall = FALSE,
                                pvalue.threshold = 0.1,adjust.method = "BH",
                                col = c("firebrick", "blue"), ylabel = "cells_per_mcL")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning in metalonda(Count = as.matrix(count), Time = time, Group =
## group, : NaNs produced

## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## MetaLonDA on Neutrophil_perc
count = meta_diversity$Neutrophil_perc
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess", points = points,
                                text = "Neutrophil_perc", parall = FALSE,
                                pvalue.threshold = 0.1, adjust.method = "BH",
                                col = c("firebrick", "blue"), ylabel = "Neutrophil_perc")
```

```
## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_path).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## MetaLonDA on Lymphocyte_perc
count = meta_diversity$Lymphocyte_perc
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess", points = points,
                                text = "Lymphocyte_perc", parall = FALSE, pvalue.threshold = 0.1,
                                adjust.method = "BH", col = c("firebrick", "blue"),
                                ylabel = "Lymphocyte_perc")
```

```
## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_path).
```

```
## MetaLonDA on RecentTacrolimusTroughLevel_ng_per_mL
count = meta_diversity$RecentTacrolimusTroughLevel_ng_per_mL
```

```
time = meta_diversity$TimeAfterTransplantation
id = meta_diversity$AnnotatedID
group = meta_diversity$Outcome
points = seq(1, 460, length.out = 100)
output.metalonda.f1 = metalonda(Count = as.matrix(count), Time = time, Group = group,
                                ID = id, n.perm = 1000, fit.method = "lowess", points = points,
                                text = "RecentTacrolimusTroughLevel_ng_per_mL", parall = FALSE,
                                pvalue.threshold = 0.1, adjust.method = "BH",
                                col = c("firebrick", "blue"),
                                ylabel = "RecentTacrolimusTroughLevel_ng_per_mL")
```

MetaLonDA on taxa

```
######################################################################
## Filteraion to limit number of tested features in MetaLonDA ######
######################################################################
### Retain taxa which appears in at leas 5% of samples with at minimum 5 reads. Based on:
level = "Species"
microbial.subset = subset_taxa(microbial_norm, !is.na(level) & !level %in% c("", "uncharacterized"))
prevdf = apply(X = otu_table(microbial.subset),
               MARGIN = ifelse(taxa_are_rows(microbial.subset), yes = 1, no = 2),
               FUN = function(x){sum(x > 5)})
# Add taxonomy and total read counts to this data.frame
prevdf = data.frame(Prevalence = prevdf,
                    TotalAbundance = taxa_sums(microbial.subset),
                    tax_table(microbial.subset))

plyr::ddply(prevdf, level, function(df1){cbind(mean(df1$Prevalence),sum(df1$Prevalence))})


# Subset to the remaining phyla
prevdf1 = subset(prevdf, Species %in% get_taxa_unique(microbial.subset, "Species"))
ggplot(prevdf1, aes(TotalAbundance, Prevalence / nsamples(microbial.subset),color=Species))+
  #Include a guess for parameter
  geom_hline(yintercept = 0.05, alpha = 0.5, linetype = 2) +  geom_point(size = 2, alpha = 0.7) +
  scale_x_log10() +  xlab("Total Abundance") + ylab("Prevalence [Frac. Samples]") +
  facet_wrap(~Species) + theme(legend.position="none")

prevalenceThreshold = 0.05 * nsamples(microbial.subset)
prevalenceThreshold
keepTaxa = rownames(prevdf1)[(prevdf1$Prevalence >= prevalenceThreshold)]
microbial.filtered = prune_taxa(keepTaxa, microbial.subset)
```

```
### Save Burkholderiaceae and Pseudomonadaceae families
Burkholderiaceae_Family = subset_taxa(microbial.filtered, Family == "Burkholderiaceae")
save(Burkholderiaceae_Family, file = "Burkholderiaceae_Family_PhyloseqObject.RData")

Pseudomonadaceae_Family = subset_taxa(microbial.filtered, Family == "Pseudomonadaceae")
save(Pseudomonadaceae_Family, file = "Pseudomonadaceae_Family_PhyloseqObject.RData")

View(otu_table(Pseudomonadaceae_Family))
View(tax_table(Pseudomonadaceae_Family))
```

```
### MetaLonDA on Phyla
## Change level to change the rank on which MetaLonDA needs to test
```

```
level =  "Phylum"
microbial.glom = tax_glom(microbial.filtered, level)
dim(otu_table(microbial.glom))

apply(otu_table(microbial.glom), 1 , sum)
apply(otu_table(microbial.glom), 1 , mean)
apply(otu_table(microbial.glom), 1 , median)

## Rename rownames to be bacteria microbial name instead of the TaxID
microbial.glom.count = as.data.frame(otu_table(microbial.glom))
for (i in rownames(microbial.glom.count))
{
  x = as.vector(tax_table(microbial)[which(rownames(tax_table(microbial)) == i), level])
  lab = sprintf("%s_%s", x, i)
  rownames(microbial.glom.count)[which(rownames(microbial.glom.count) == i)] = lab
  cat(i, "\n")
}


Group_real = as.vector(as.data.frame(sample_data(microbial.glom))$Outcome)
ID_real = as.vector(as.data.frame(sample_data(microbial.glom))$AnnotatedID)
Time_real = as.data.frame(sample_data(microbial.glom))$TimeAfterTransplantation

output_all_nbinomial_phylum = metalondaAll(Count = microbial.glom.count[c(2,3,6,7,12),],
                                           Time = Time_real, Group = Group_real,ID = ID_real,
                                           fit.method = "nbinomial", n.perm = 1000,
                                           num.intervals = 99, parall = FALSE,
                                           pvalue.threshold = 0.1, adjust.method = "BH",
                                           time.unit = "days", norm.method = "none",
                                           prefix = "LTx_Phylum", col = c("firebrick", "blue"))
```

```
### MetaLonDA on class level
## Change level to change the rank on which MetaLonDA needs to test
level =  "Class"
microbial.glom = tax_glom(microbial.filtered, level)
dim(otu_table(microbial.glom))

apply(otu_table(microbial.glom), 1 , sum)
apply(otu_table(microbial.glom), 1 , mean)
apply(otu_table(microbial.glom), 1 , median)

## Rename rownames to be bacteria microbial name instead of the TaxID
microbial.glom.count = as.data.frame(otu_table(microbial.glom))
for (i in rownames(microbial.glom.count))
{
  x = as.vector(tax_table(microbial)[which(rownames(tax_table(microbial)) == i), level])
  lab = sprintf("%s_%s", x, i)
  rownames(microbial.glom.count)[which(rownames(microbial.glom.count) == i)] = lab
  cat(i, "\n")
}


Group_real = as.vector(as.data.frame(sample_data(microbial.glom))$Outcome)
ID_real = as.vector(as.data.frame(sample_data(microbial.glom))$AnnotatedID)
```

```
Time_real = as.data.frame(sample_data(microbial.glom))$TimeAfterTransplantation

output_all_nbinomial_class = metalondaAll(Count = microbial.glom.count, Time = Time_real,
                                          Group = Group_real, ID = ID_real,
                                          fit.method = "nbinomial", n.perm = 1000,
                                          num.intervals = 99, parall = FALSE,
                                          pvalue.threshold = 0.1, adjust.method = "BH",
                                          time.unit = "days", norm.method = "none",
                                          prefix = "LTx_class", col = c("firebrick", "blue"))
```

```
### MetaLonDA on order level
## Change level to change the rank on which MetaLonDA needs to test
level =  "Order"
microbial.glom = tax_glom(microbial.filtered, level)
dim(otu_table(microbial.glom))

apply(otu_table(microbial.glom), 1 , sum)
apply(otu_table(microbial.glom), 1 , mean)
apply(otu_table(microbial.glom), 1 , median)

## Rename rownames to be bacteria microbial name instead of the TaxID
microbial.glom.count = as.data.frame(otu_table(microbial.glom))
for (i in rownames(microbial.glom.count))
{
  x = as.vector(tax_table(microbial)[which(rownames(tax_table(microbial)) == i), level])
  lab = sprintf("%s_%s", x, i)
  rownames(microbial.glom.count)[which(rownames(microbial.glom.count) == i)] = lab
  cat(i, "\n")
}

Group_real = as.vector(as.data.frame(sample_data(microbial.glom))$Outcome)
ID_real = as.vector(as.data.frame(sample_data(microbial.glom))$AnnotatedID)
Time_real = as.data.frame(sample_data(microbial.glom))$TimeAfterTransplantation

output_all_nbinomial_order= metalondaAll(Count = microbial.glom.count, Time = Time_real,
                                         Group = Group_real, ID = ID_real,
                                         fit.method = "nbinomial", n.perm = 1000,
                                         num.intervals = 99, parall = FALSE,
                                         pvalue.threshold = 0.1, adjust.method = "BH",
                                         time.unit = "days", norm.method = "none",
                                         prefix = "LTx_order", col = c("firebrick", "blue"))
```

```
### MetaLonDA on order level
## Change level to change the rank on which MetaLonDA needs to test
level =  "Family"
microbial.glom = tax_glom(microbial.filtered, level)
dim(otu_table(microbial.glom))

apply(otu_table(microbial.glom), 1 , sum)
apply(otu_table(microbial.glom), 1 , mean)
apply(otu_table(microbial.glom), 1 , median)

## Rename rownames to be bacteria microbial name instead of the TaxID
microbial.glom.count = as.data.frame(otu_table(microbial.glom))
```

```
for (i in rownames(microbial.glom.count))
{
  x = as.vector(tax_table(microbial)[which(rownames(tax_table(microbial)) == i), level])
  lab = sprintf("%s_%s", x, i)
  rownames(microbial.glom.count)[which(rownames(microbial.glom.count) == i)] = lab
  cat(i, "\n")
}

Group_real = as.vector(as.data.frame(sample_data(microbial.glom))$Outcome)
ID_real = as.vector(as.data.frame(sample_data(microbial.glom))$AnnotatedID)
Time_real = as.data.frame(sample_data(microbial.glom))$TimeAfterTransplantation

output_all_nbinomial_family = metalondaAll(Count = microbial.glom.count, Time = Time_real,
                                           Group = Group_real, ID = ID_real,
                                           fit.method = "nbinomial", n.perm = 1000,
                                           num.intervals = 99, parall = FALSE,
                                           pvalue.threshold = 0.1, adjust.method = "BH",
                                           time.unit = "days", norm.method = "none",
                                          prefix = "LTx_family", col = c("firebrick", "blue"))
```

```
########################################
########## MetaLonDA for Burlkerderia (all ranks) ##
########################################

Burk_Order = subset_taxa(microbial.filtered, Order == "Burkholderiales")
############### Agglomerate Taxa ####################

level =  "Order"
Burk_Order.glom = tax_glom(Burk_Order, level)
dim(otu_table(Burk_Order.glom))


level =  "Family"
Burk_Order.glom_family = tax_glom(Burk_Order, level)
dim(otu_table(Burk_Order.glom_family))


level =  "Genus"
Burk_Order.glom_genus = tax_glom(Burk_Order, level)
dim(otu_table(Burk_Order.glom_genus))

level =  "Species"
Burk_Order.glom_spices = tax_glom(Burk_Order, level)
dim(otu_table(Burk_Order.glom_spices))


### MetaLonDA for all burckolderia (genus) species
## Rename rownames to be bacteria microbial name instead of the TaxID
## TODO: replace "Burk_Order.glom_spices" argument with the rank that wanted to be tested
microbial.glom = Burk_Order.glom_spices ## TODO: Change this if you want to change the rank
level = "Species"  ### TODO: Change this if you want to change the rank
microbial.glom.count = as.data.frame(otu_table(microbial.glom))
for (i in rownames(microbial.glom.count))
{
  x = as.vector(tax_table(microbial)[which(rownames(tax_table(microbial)) == i), level])
  lab = sprintf("%s_%s", x, i)
  rownames(microbial.glom.count)[which(rownames(microbial.glom.count) == i)] = lab
  cat(i, "\n")
}

Group_real = as.vector(as.data.frame(sample_data(microbial.glom))$Outcome)
ID_real = as.vector(as.data.frame(sample_data(microbial.glom))$AnnotatedID)
Time_real = as.data.frame(sample_data(microbial.glom))$TimeAfterTransplantation

output_all_nbinomial_burk_species = metalondaAll(Count = microbial.glom.count,
                                                 Time = Time_real, Group = Group_real,
                                                ID = ID_real, fit.method = "nbinomial",
                                                n.perm = 1000, num.intervals = 99,
                                                parall = FALSE, pvalue.threshold = 0.1,
                                                adjust.method = "BH", time.unit = "days",
                                                norm.method = "none", prefix = "Lx_burk_species",
                                                col = c("firebrick", "blue"))
```