botero_analysis

Sean Lu

2025-06-10

Setup

```
library("phyloseq")
library("ggplot2")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tibble")
library("ggpubr")
library("phylosmith")
```

```
## Registered S3 method overwritten by 'dendextend':
##   method     from
##   rev.hclust vegan
```

```
library("DESeq2")
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:phyloseq':
##
##     distance

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment
```

```
## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
## The following object is masked from 'package:phyloseq':
##
##     sampleNames
```

```
library("EnhancedVolcano")
```

```
## Loading required package: ggrepel
```

```
library("microbiome")
```

```
##
## microbiome R package (microbiome.github.com)
##
##
##
##  Copyright (C) 2011-2022 Leo Lahti,
##     Sudarshan Shetty et al. <microbiome.github.io>

##
## Attaching package: 'microbiome'

## The following object is masked from 'package:SummarizedExperiment':
##
##     coverage

## The following object is masked from 'package:GenomicRanges':
##
##     coverage

## The following objects are masked from 'package:IRanges':
##
##     coverage, transform

## The following object is masked from 'package:S4Vectors':
##
##     transform

## The following object is masked from 'package:ggplot2':
##
##     alpha

## The following object is masked from 'package:base':
##
##     transform
```

```
library("eulerr")
library("ggVennDiagram")
```

```
##
## Attaching package: 'ggVennDiagram'

## The following object is masked from 'package:microbiome':
##
##     overlap
```

Load in processed Data

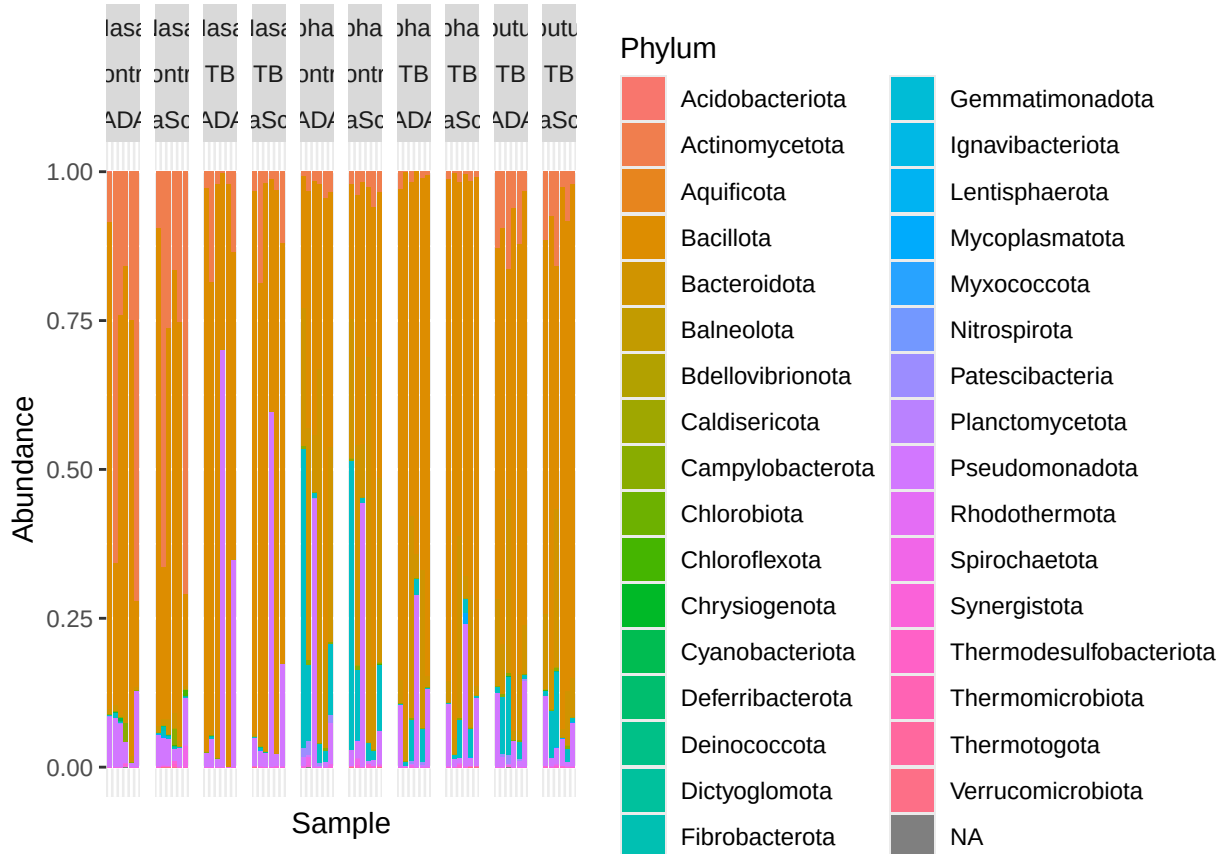
```
ps_dada2 <- readRDS("data_processed/botero_2014/ps_dada2.rds")
ps_ms <- readRDS("data_processed/botero_2014/ps_metascope_priors_trimmed.rds")

ps_dada2_oral <- subset_samples(ps_dada2, Sample_type == "Oropharynx")
ps_ms_oral <- subset_samples(ps_ms, Sample_type == "Oropharynx")
ps_dada2_nasal <- subset_samples(ps_dada2, Sample_type == "Nasal")
ps_ms_nasal <- subset_samples(ps_ms, Sample_type == "Nasal")


dada2_df <- psmelt(ps_dada2) |>
  dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species)))
dada2_df$pipeline = "DADA2"

ms_df <- psmelt(ps_ms)
ms_df$pipeline = "MetaScope"
ms_df$kingdom = "Bacteria"
ms_df <- ms_df |>
  dplyr::relocate(kingdom, .before = phylum)
colnames(ms_df) <- c("OTU", "Sample", "Abundance", "Sequencing_Type", "Patient",
                     "Sample_type", "status", "Kingdom", "Phylum", "Class",
                     "Order", "Family", "Genus", "Species", "pipeline")

merged_df <- rbind(dada2_df, ms_df)

dada2_relab_df <- phylosmith::relative_abundance(ps_dada2) |>
  phyloseq::psmelt() |>
  dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species)))
dada2_relab_df$pipeline = "DADA2"

ms_relab_df <- phylosmith::relative_abundance(ps_ms) |>
  phyloseq::psmelt()
ms_relab_df$pipeline = "MetaScope"
ms_relab_df$kingdom = "Bacteria"
ms_relab_df <- ms_relab_df |>
  dplyr::relocate(kingdom, .before = phylum)
colnames(ms_relab_df) <- c("OTU", "Sample", "Abundance", "Sequencing_Type", "Patient",
                     "Sample_type", "status", "Kingdom", "Phylum", "Class",
                     "Order", "Family", "Genus", "Species", "pipeline")

merged_relab_df <- rbind(dada2_relab_df, ms_relab_df)
```

The DADA2 data is generated from the `dada2_botero.Rmd` file. The MetaScope data was generated from the `process_metascope_id.R` functions.

Plotting relative abundances of MetaScope and DADA2

Phylum Level Abundances

```
top_phyla <- merged_relab_df |>
  dplyr::group_by(Phylum) |>
  dplyr::summarise(total_abund = sum(Abundance), .groups = "drop") |>
  dplyr::filter(!is.na(Phylum)) |>
  slice_max(order_by=total_abund, n = 5) |>
  pull(Phylum) |>
  sort()

relab_phylum <- ggplot(merged_df,
                       aes(x = Sample, y = Abundance, fill = Phylum)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status, pipeline), scales = "free_x") +
  scale_fill_discrete() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
relab_phylum
```

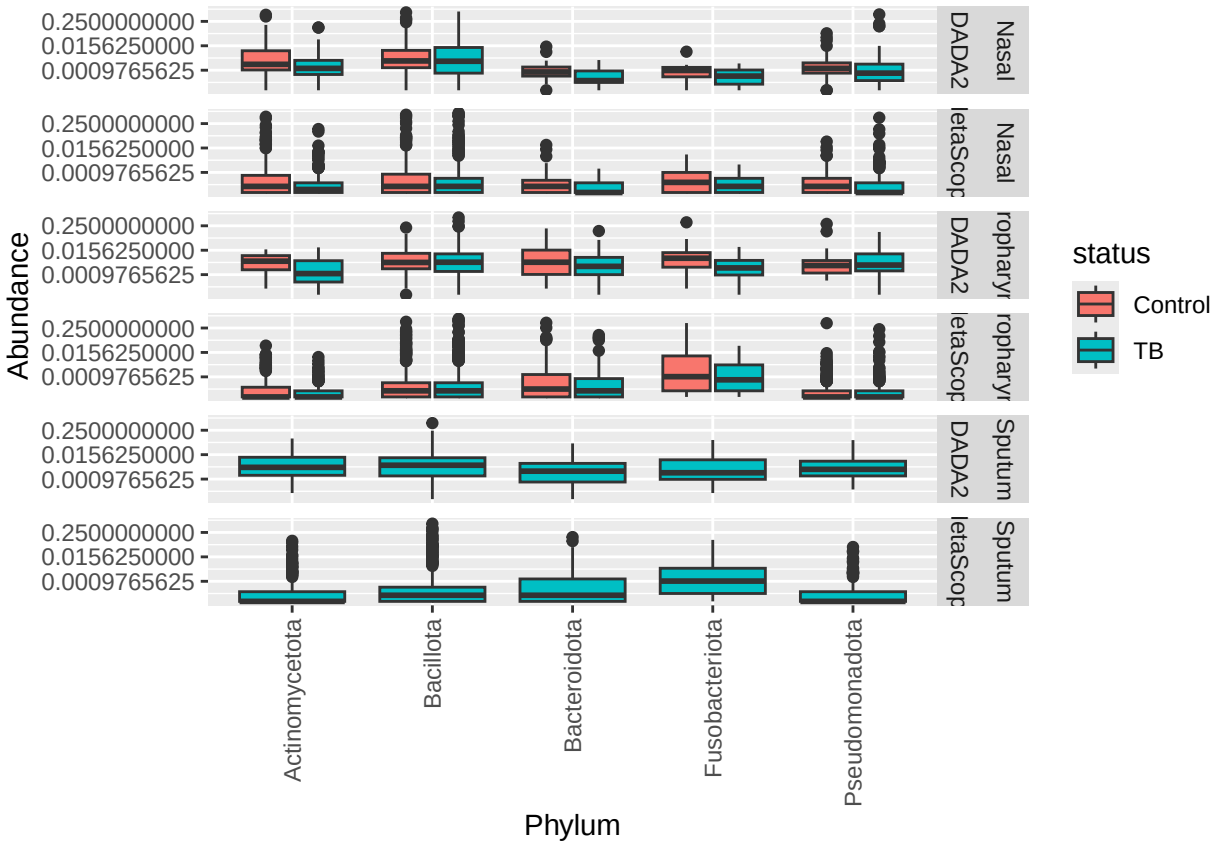



```
relab_phylum_legend <- get_legend(relab_phylum)

merged_relab_df |>
  dplyr::filter(Phylum %in% top_phyla) |>
  ggplot(aes(fill=status, y=Abundance, x=Phylum)) +
  geom_boxplot() +
  facet_grid(vars(Sample_type, pipeline)) +
```

```
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_y_continuous(trans='log2')
```

```
## Warning in scale_y_continuous(trans = "log2"): log-2 transformation introduced
## infinite values.
```

```
## Warning: Removed 134440 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

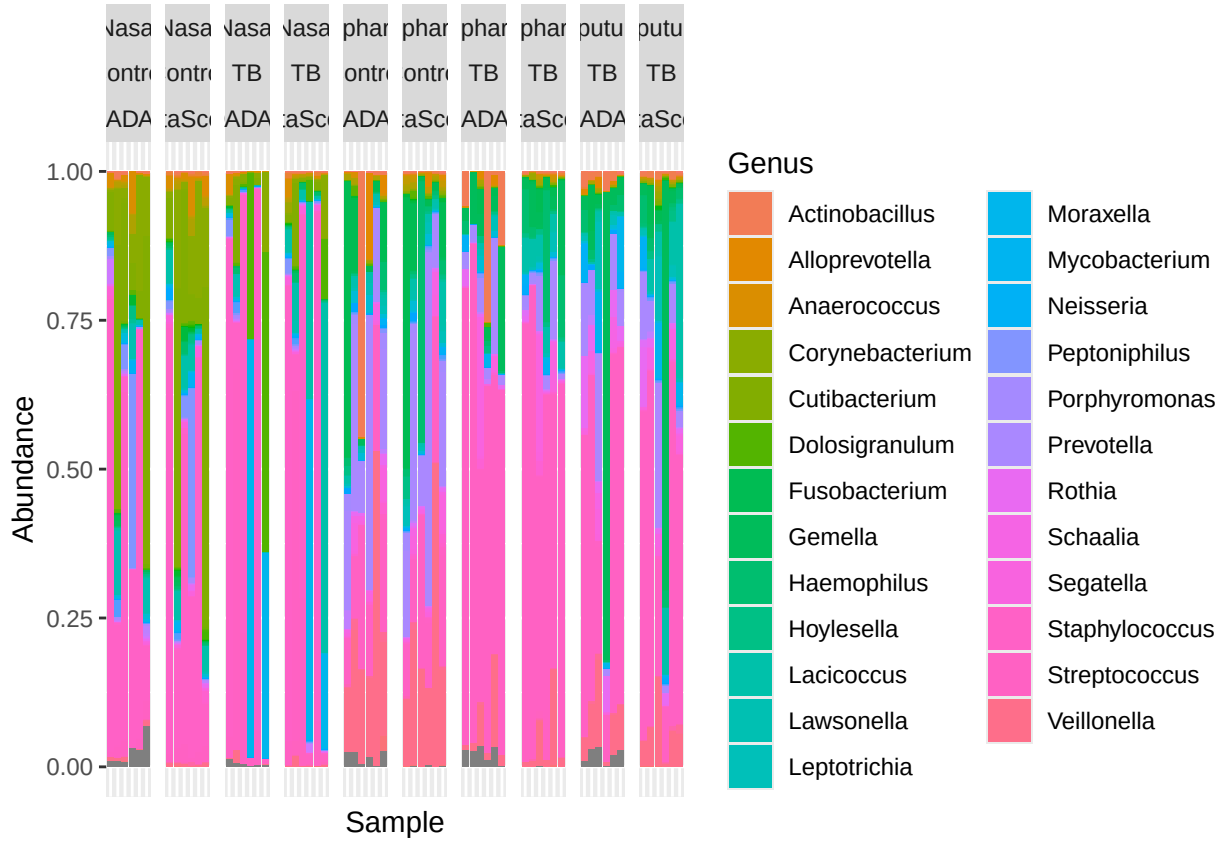


At Phylum level taxonomies, both DADA2 and MetaScope show similar results. Nasal samples in the controls show increased abundance of Acidobacteriota compared to TB sample and a decreased relative abundance in Pseudomonadota in controls relative to TB samples. The oropharynx samples mild decrease in Baciollota phyla and increases in Fusobacteriota and Pseudomonadota in the controls compared to TB positive samples.

Genus Level Abundances

```
top_genera <- merged_relab_df |>
  dplyr::group_by(Genus) |>
  dplyr::summarise(total_abund = sum(Abundance), .groups = "drop") |>
  dplyr::filter(!is.na(Genus)) |>
  slice_max(order_by=total_abund, n = 25) |>
  pull(Genus) |>
  sort()
```

```
relab_genus <- ggplot(merged_df,
                      aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status, pipeline), scales = "free_x") +
  scale_fill_discrete(breaks = top_genera) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
relab_genus
```

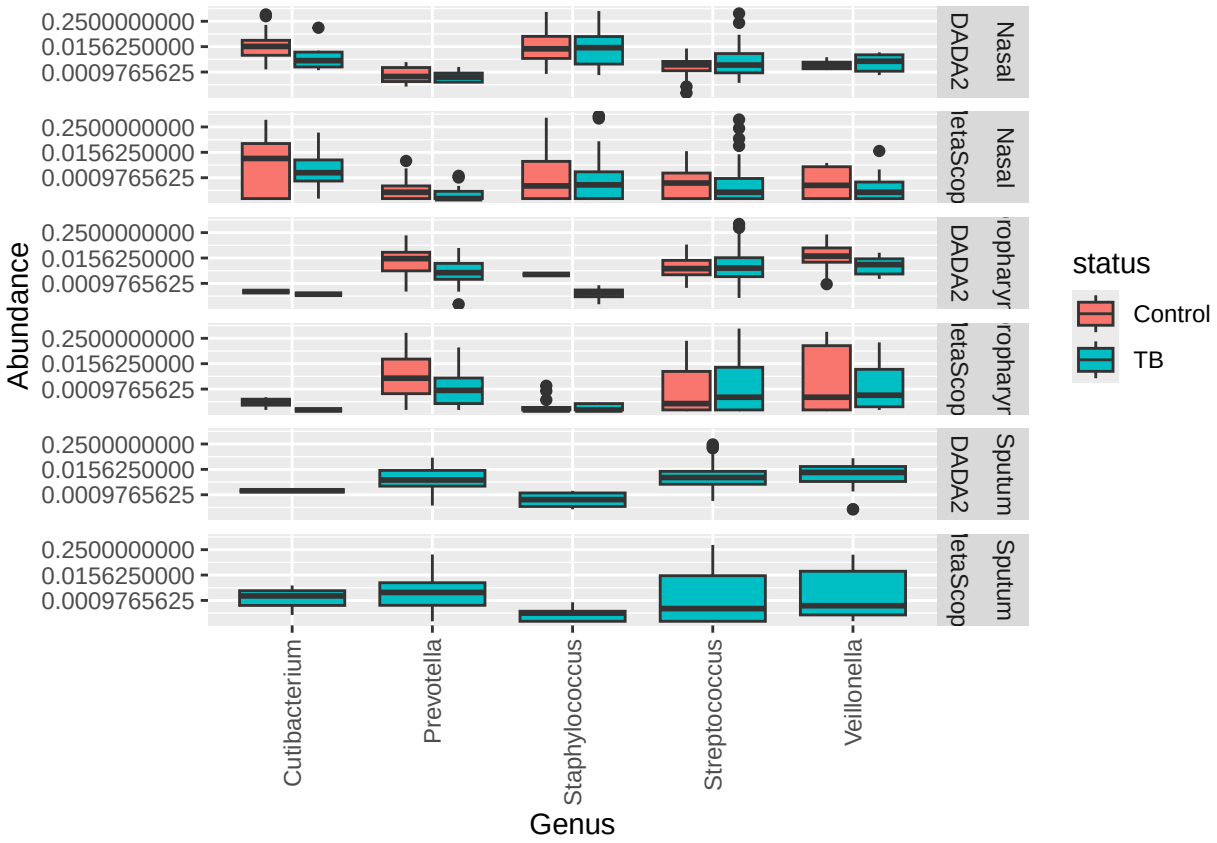



```
relab_genus_legend <- get_legend(relab_genus)

top_genera <- merged_relab_df |>
  dplyr::group_by(Genus) |>
  dplyr::summarise(total_abund = sum(Abundance), .groups = "drop") |>
  dplyr::filter(!is.na(Genus)) |>
  slice_max(order_by=total_abund, n = 5) |>
  pull(Genus) |>
  sort()

merged_relab_df |>
  dplyr::filter(Genus %in% top_genera) |>
  ggplot(aes(fill=status, y=Abundance, x=Genus)) +
  geom_boxplot() +
  facet_grid(vars(Sample_type, pipeline)) +
```

```
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_y_continuous(trans='log2')
```

```
## Warning in scale_y_continuous(trans = "log2"): log-2 transformation introduced
## infinite values.
```

```
## Warning: Removed 27783 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

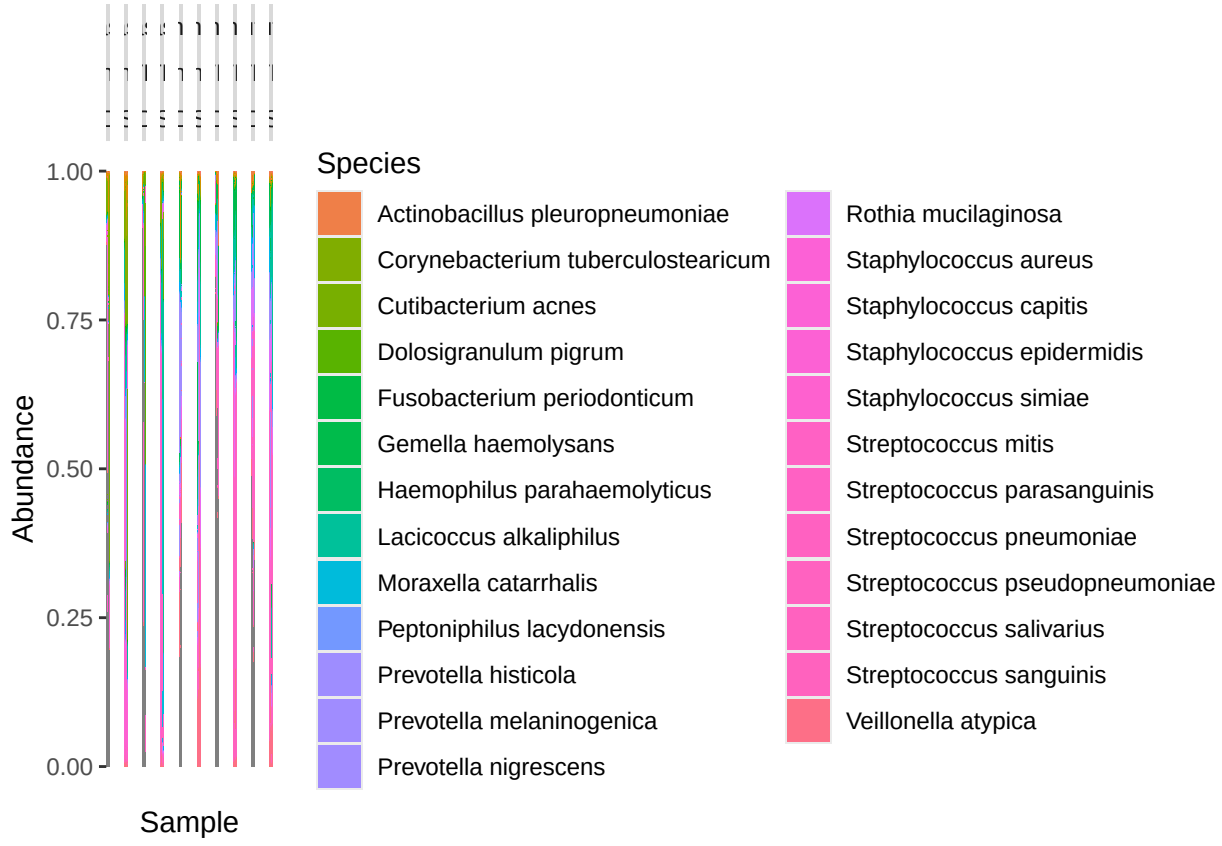


At the genus level, still DADA2 and MetaScope have similar relative abundances and identify the same genera that are differentially expressed. Notably, DADA2 identifies more Actinobacillus

Species Level Abundances

```
top_species <- merged_relab_df |>
  dplyr::group_by(Species, Genus) |>
  dplyr::summarise(total_abund = sum(Abundance), .groups = "drop") |>
  dplyr::filter(!is.na(Species)) |>
  slice_max(order_by=total_abund, n = 25) |>
  pull(Species) |>
  sort()

relab_species <- ggplot(merged_df,
```

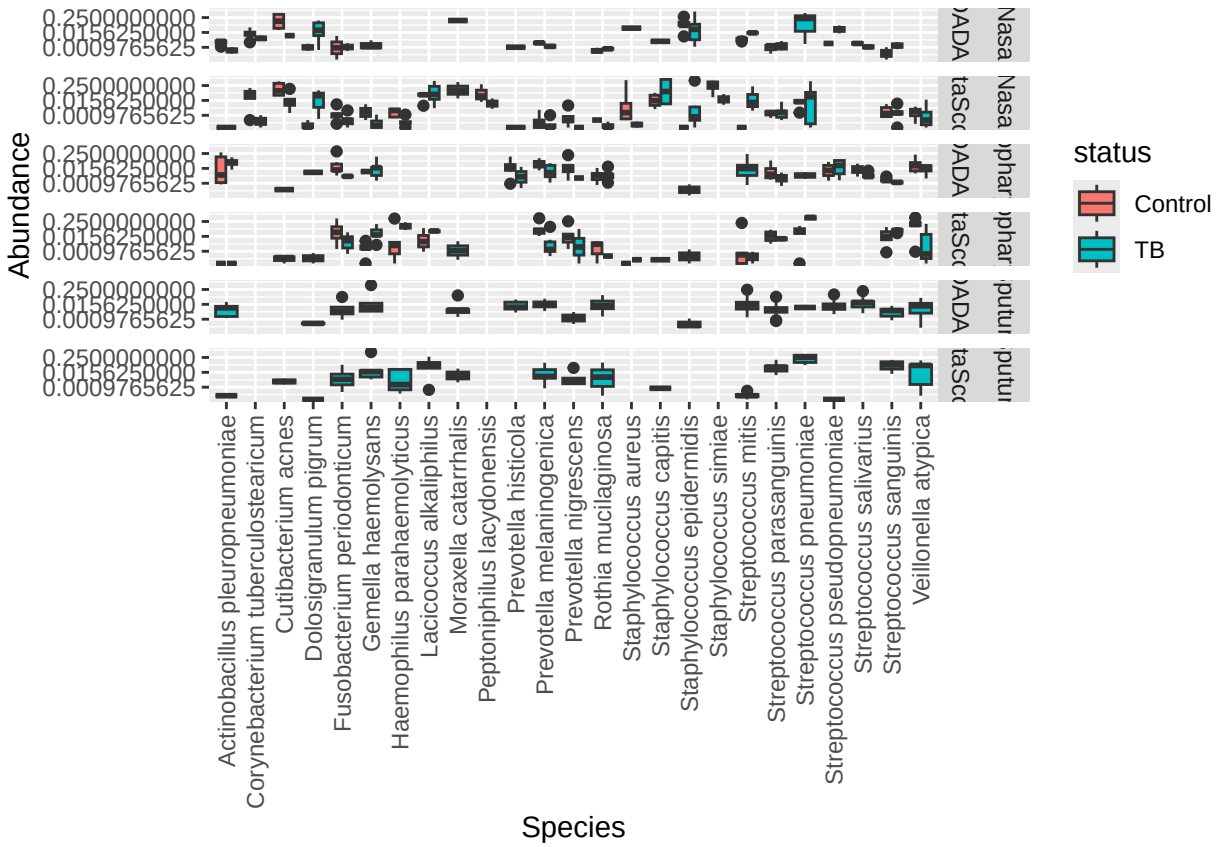
```
                        aes(x = Sample, y = Abundance, fill = Species)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status, pipeline), scales = "free_x") +
  scale_fill_discrete(breaks = top_species) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
relab_species
```




```
relab_species_legend <- get_legend(relab_species)

merged_relab_df |>
  dplyr::filter(Species %in% top_species) |>
  ggplot(aes(fill=status, y=Abundance, x=Species)) +
  geom_boxplot() +
  facet_grid(vars(Sample_type, pipeline)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_y_continuous(trans='log2')
```

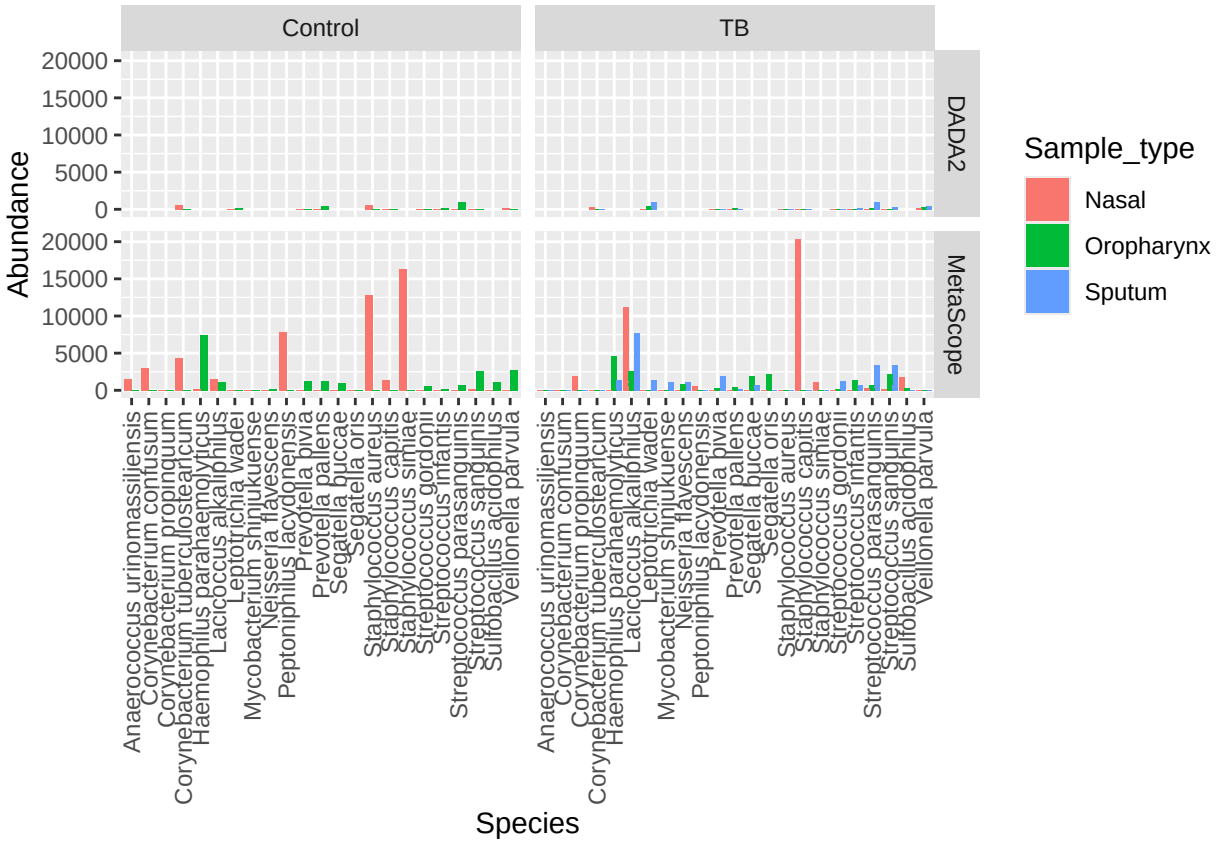
```
## Warning in scale_y_continuous(trans = "log2"): log-2 transformation introduced
## infinite values.
```

```
## Warning: Removed 12508 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

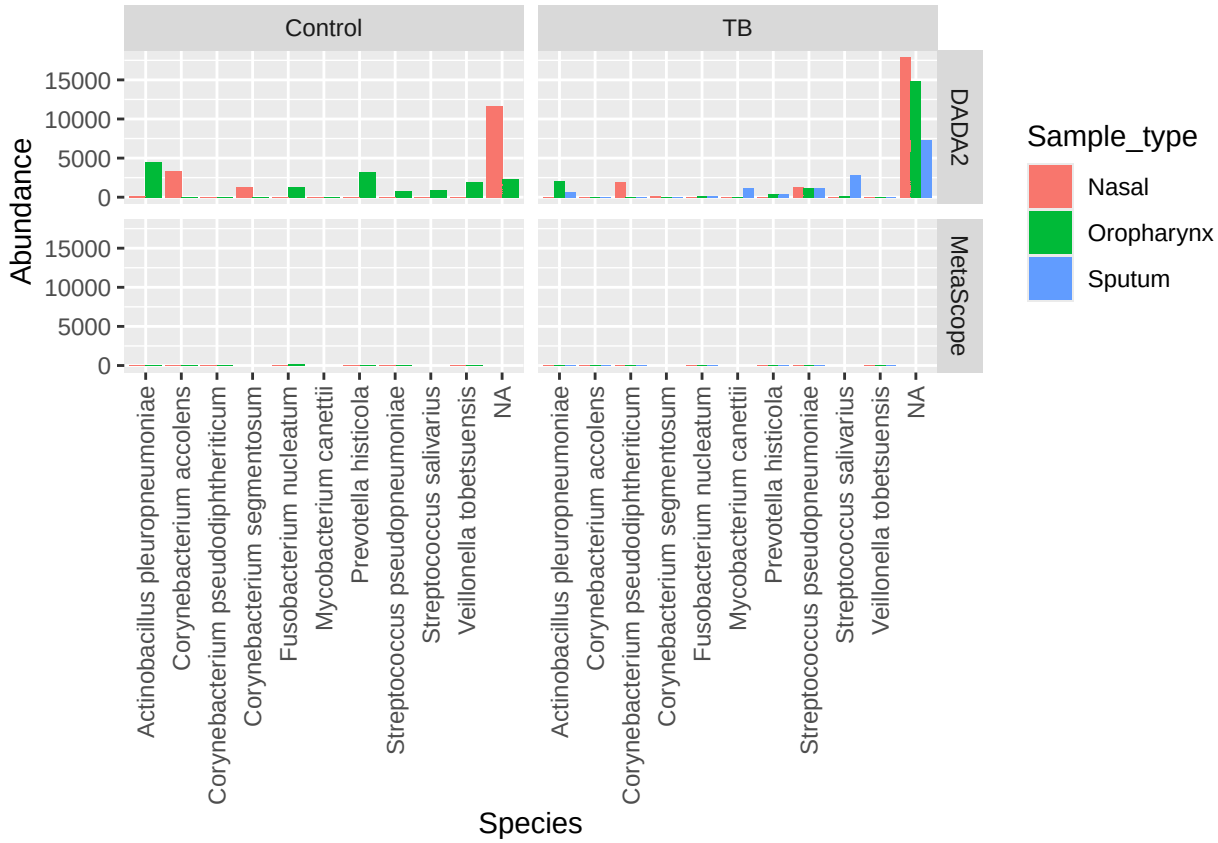


Filtered abundance barplots

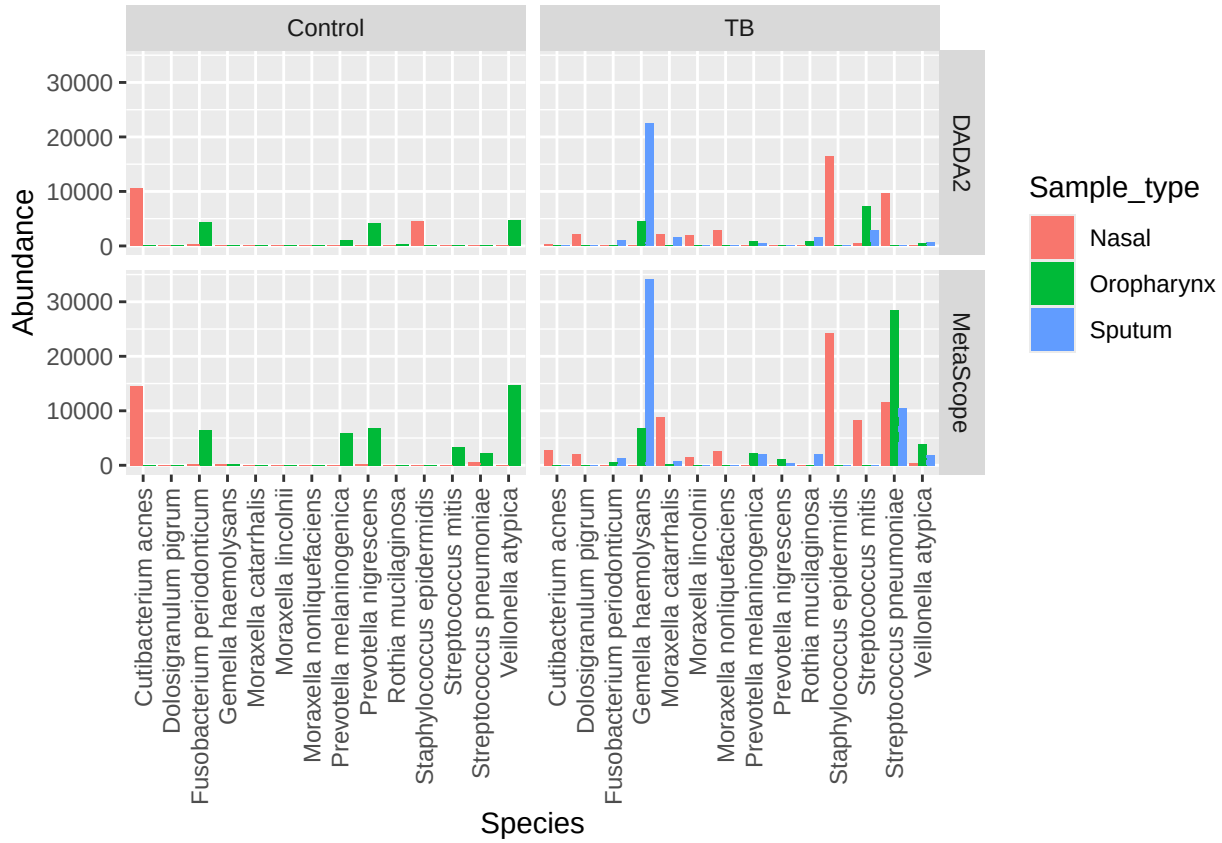
```
high_abund_dada2 <- merged_df |>
  dplyr::filter(Abundance > 1000) |>
  dplyr::filter(pipeline == "DADA2")

high_abund_ms <- merged_df |>
  dplyr::filter(Abundance > 1000) |>
  dplyr::filter(pipeline == "MetaScope")

select_species_both <- unique(high_abund_dada2$Species[high_abund_dada2$Species %in% high_abund_ms$Spec
select_species_dada2 <- unique(high_abund_dada2$Species[!(high_abund_dada2$Species %in% high_abund_ms$Sp
select_species_ms <- unique(high_abund_ms$Species[!(high_abund_ms$Species %in% high_abund_dada2$Species

ms_unique_species <- merged_df |>
  dplyr::filter(Species %in% select_species_ms) |>
  ggplot(aes(fill=Sample_type, y=Abundance, x=Species)) +
  geom_bar(position="dodge", stat="identity") +
  facet_grid(vars(pipeline), vars(status)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
ms_unique_species
```




```
dada2_unique_species <- merged_df |>
  dplyr::filter(Species %in% select_species_dada2) |>
  ggplot(aes(fill=Sample_type, y=Abundance, x=Species)) +
  geom_bar(position="dodge", stat="identity") +
  facet_grid(vars(pipeline), vars(status)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
dada2_unique_species
```

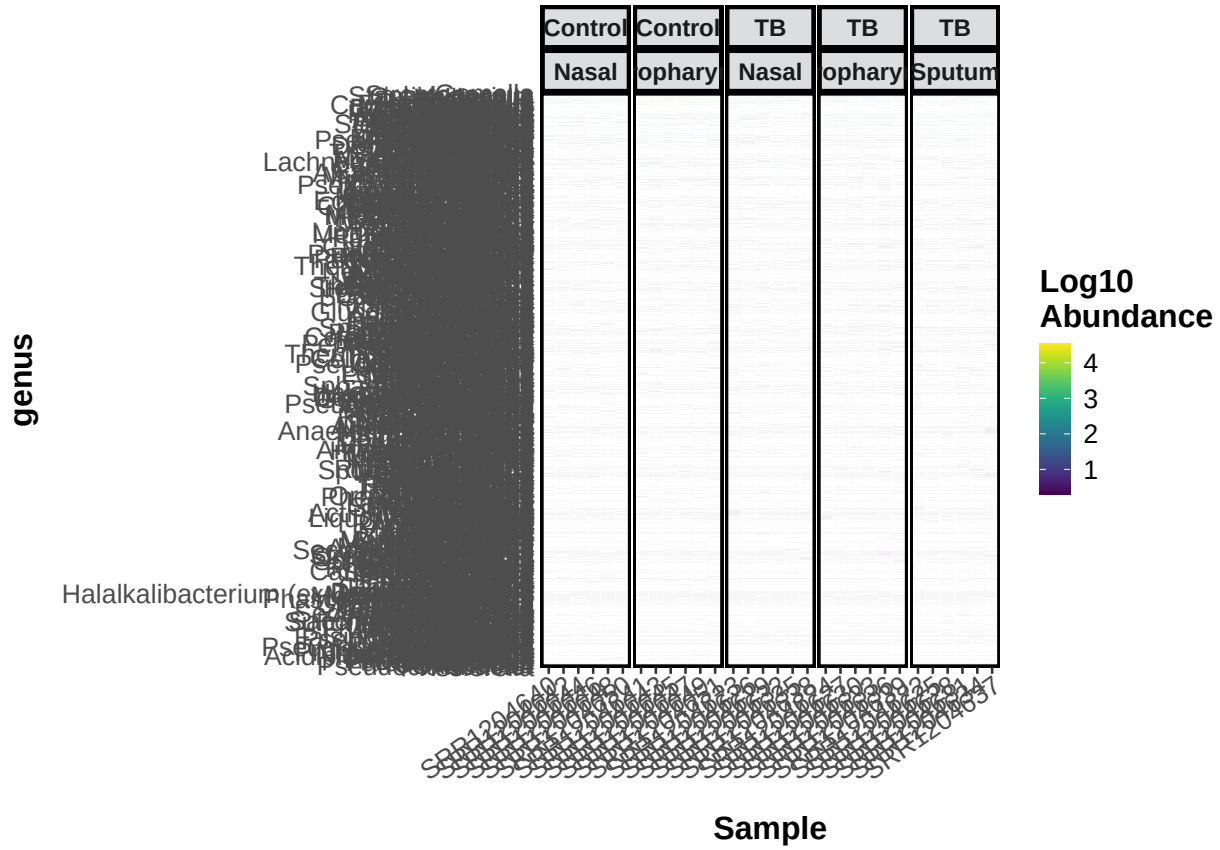



```
high_abund_species <- merged_df |>
  dplyr::filter(Species %in% select_species_both) |>
  ggplot(aes(fill=Sample_type, y=Abundance, x=Species)) +
  geom_bar(position="dodge", stat="identity") +
  facet_grid(vars(pipeline), vars(status)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
high_abund_species
```

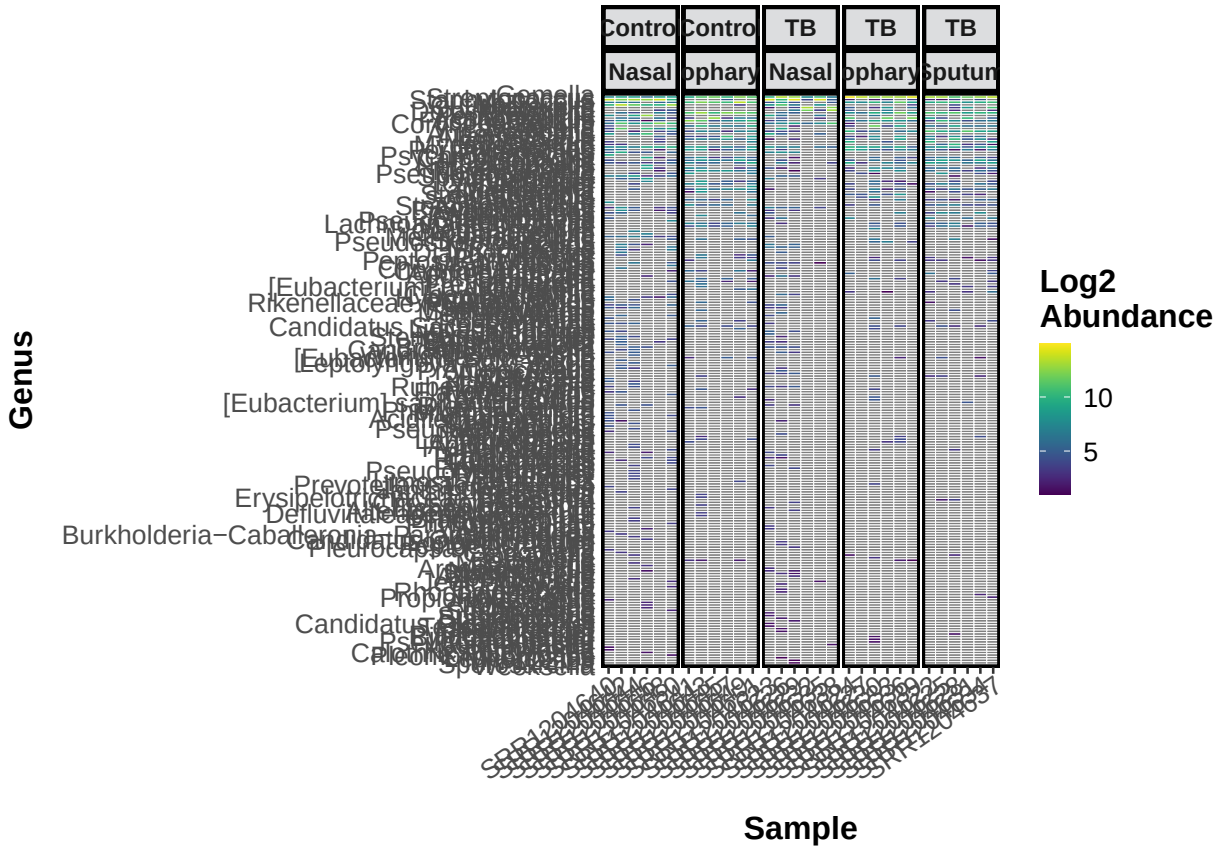


Heatmaps

```
ps_ms_filt <- taxa_filter(ps_ms, frequency = 0.05)

abundance_heatmap(ps_ms_filt, classification = 'genus',
  treatment = "Sample_type", transformation = 'log10') +
  facet_wrap(vars(status,Sample_type), nrow = 1, scales = "free_x")
```




```
ps_dada2_filt <- taxa_filter(ps_dada2, frequency = 0.01)
abundance_heatmap(ps_dada2_filt, classification = 'Genus',
  treatment = "Sample_type", transformation = 'log2') +
  facet_wrap(vars(status,Sample_type), nrow = 1, scales = "free_x")
```



Plotting Alpha Diversity

```
p2_1 <- plot_richness(ps_dada2_oral, measures=c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvSi
  stat_compare_means(label = "p.signif", label.x = 1.5, comparisons = list(c("Control", "TB")),
                     method = "wilcox.test") +
  geom_boxplot()
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided c
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
p2_2 <- plot_richness(ps_ms_oral, measures=c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvSimps
  stat_compare_means(label = "p.signif", label.x = 1.5, comparisons = list(c("Control", "TB")),
                     method = "wilcox.test") +
  geom_boxplot()

p2_3 <- plot_richness(ps_dada2_nasal, measures=c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvS
  stat_compare_means(label = "p.signif", label.x = 1.5, comparisons = list(c("Control", "TB")), method =
  geom_boxplot()
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
p2_4 <- plot_richness(ps_ms_nasal, measures=c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvSimp
  stat_compare_means(label = "p.signif", label.x = 1.5, comparisons = list(c("Control", "TB")),method =
  geom_boxplot()

ggarrange(p2_1, p2_2, p2_3, p2_4, labels = "AUTO")
```

```
## Warning in wilcox.test.default(c(72, 128, 54, 60, 72, 106), c(106, 38, 111, :
## cannot compute exact p-value with ties
```

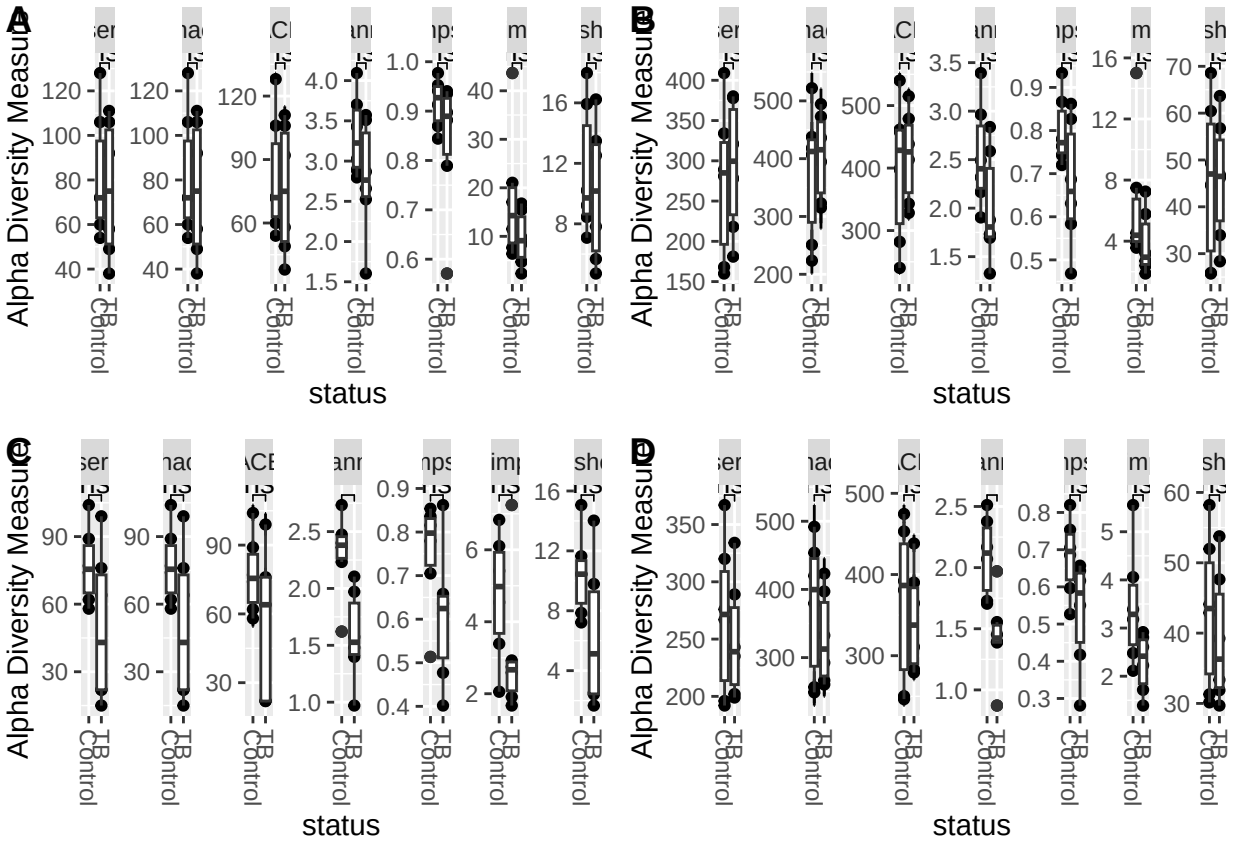
```
## Warning in wilcox.test.default(c(72, 128, 54, 60, 72, 106), c(106, 38, 111, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(c(72, 128, 54, 60, 72, 106), c(106, 38, 111, :
## cannot compute exact p-value with ties
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_signif()`).
```

```
## Warning in wilcox.test.default(c(77, 89, 104, 62, 58, 74), c(76, 99, 64, :
## cannot compute exact p-value with ties
## Warning in wilcox.test.default(c(77, 89, 104, 62, 58, 74), c(76, 99, 64, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(c(77, 89, 104, 62, 58, 74), c(76, 99, 64, :
## cannot compute exact p-value with ties
```

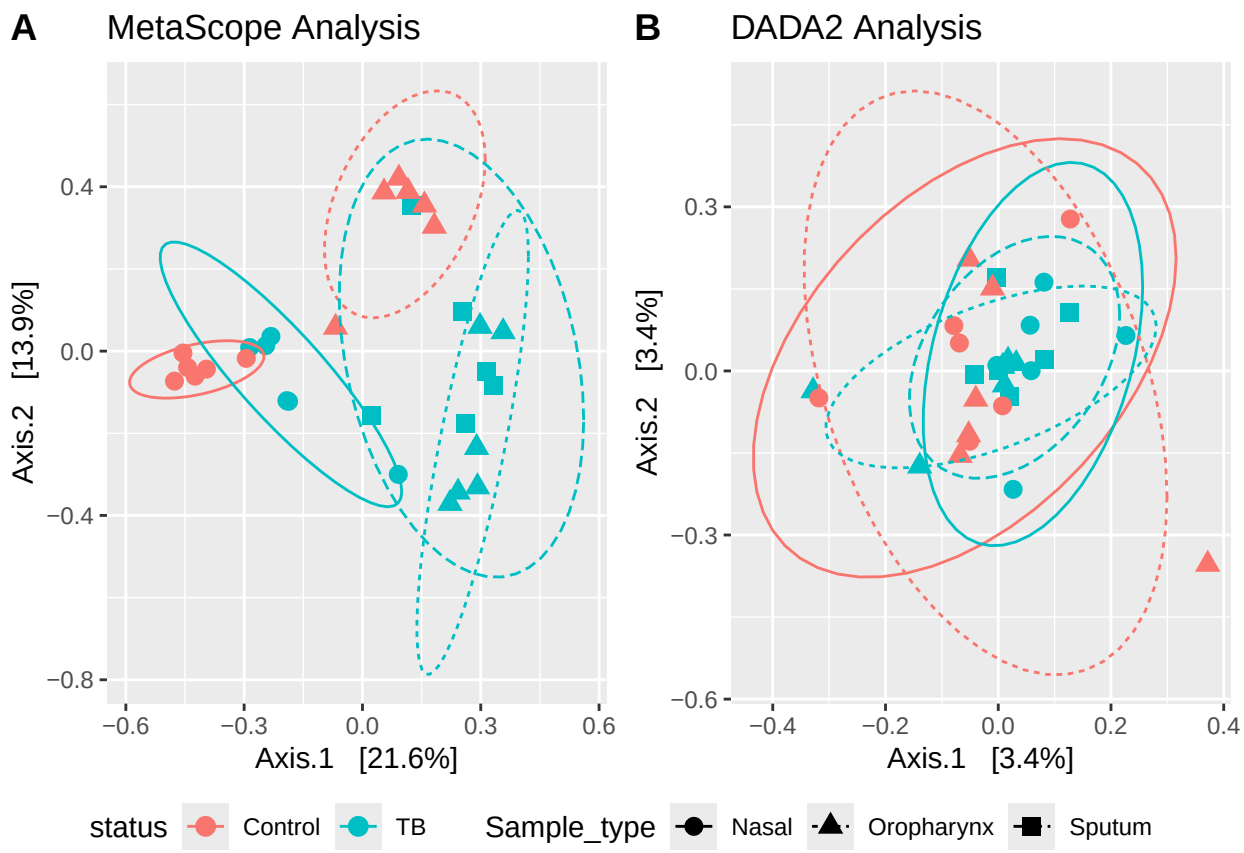
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



PCOA Plots

```
ps_ms.ord <- ordinate(ps_ms, "PCoA", "bray")
p3_1 <- plot_ordination(ps_ms, ps_ms.ord, type="samples", color="status", shape="Sample_type", title="M
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))


ps_dada2.ord <- ordinate(ps_dada2, "PCoA", "bray")
p3_2 <- plot_ordination(ps_dada2, ps_dada2.ord, type="samples", color="status", shape="Sample_type", ti
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))
ggarrange(p3_1, p3_2, labels = "AUTO", common.legend = TRUE, legend = "bottom")
```

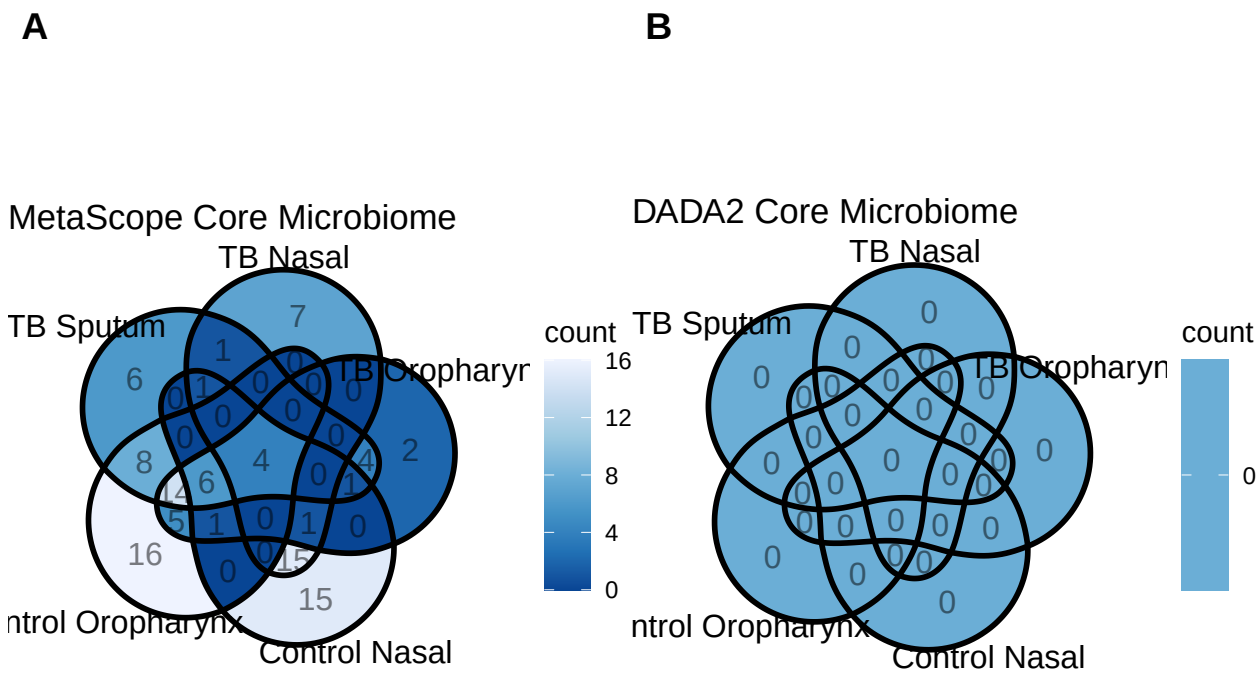
```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```



Core Microbiome Analysis

```
ps_dada2_rel <- microbiome::transform(ps_dada2, "compositional")
ps_ms_rel <- microbiome::transform(ps_ms, "compositional")

list_core_ms <- c()

groups <- list(c("TB", "Sputum"), c("TB", "Nasal"), c("TB", "Oropharynx"),
               c("Control", "Nasal"), c("Control", "Oropharynx"))
for (i in 1:5){
  ps.sub <- subset_samples(ps_ms_rel, status == groups[[i]][1] & Sample_type == groups[[i]][2])
  core_m <- core_members(ps.sub,
                         detection = 0.001,
                         prevalence = 0.2)
  list_core_ms[[i]] <- core_m
}
names(list_core_ms) <- lapply(groups, function(x) paste(x, collapse = " "))
p4_1 <- ggVennDiagram(list_core_ms, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2)) +
  ggtitle("MetaScope Core Microbiome")

list_core_dada <- c() # an empty object to store information
for (i in 1:5){
```

```
  ps.sub <- subset_samples(ps_dada2_rel, status == groups[[i]][1] & Sample_type == groups[[i]][2])
  core_m <- core_members(ps.sub,
                         detection = 0.001,
                         prevalence = 0.2)
  list_core_dada[[i]] <- core_m
}
names(list_core_dada) <- lapply(groups, function(x) paste(x, collapse = " "))
p4_2 <- ggVennDiagram(list_core_dada, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2)) +
  ggtitle("DADA2 Core Microbiome")

ggarrange(p4_1, p4_2, labels = "AUTO")
```




```
deseq_ms_oral = phyloseq_to_deseq2(ps_ms_oral, ~ status)
```

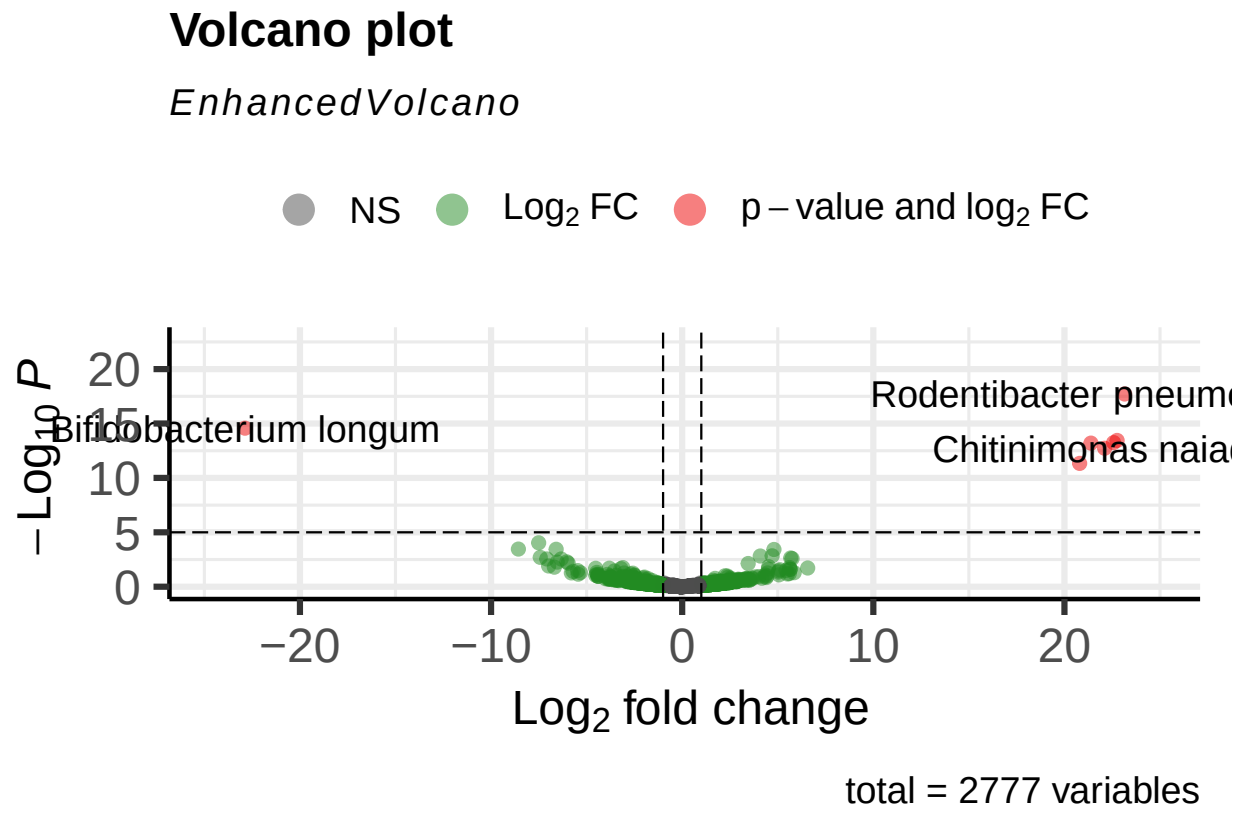
```
## converting counts to integer mode
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
deseq_ms_oral = estimateSizeFactors(deseq_ms_oral, type = 'poscounts')
deseq_ms_oral = DESeq(deseq_ms_oral, test="Wald", fitType="parametric")
```

```
## using pre-existing size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
deseq_ms_oral_res = results(deseq_ms_oral, cooksCutoff = FALSE)
deseq_ms_oral_res <- cbind(deseq_ms_oral_res,
                           as(tax_table(ps_ms_oral)[rownames(deseq_ms_oral_res), ], "matrix"))

EnhancedVolcano(deseq_ms_oral_res,
  lab = deseq_ms_oral_res$species,
  x = 'log2FoldChange',
  y = 'pvalue')
```




```
deseq_ms_nasal = phyloseq_to_deseq2(ps_ms_nasal, ~ status)
```

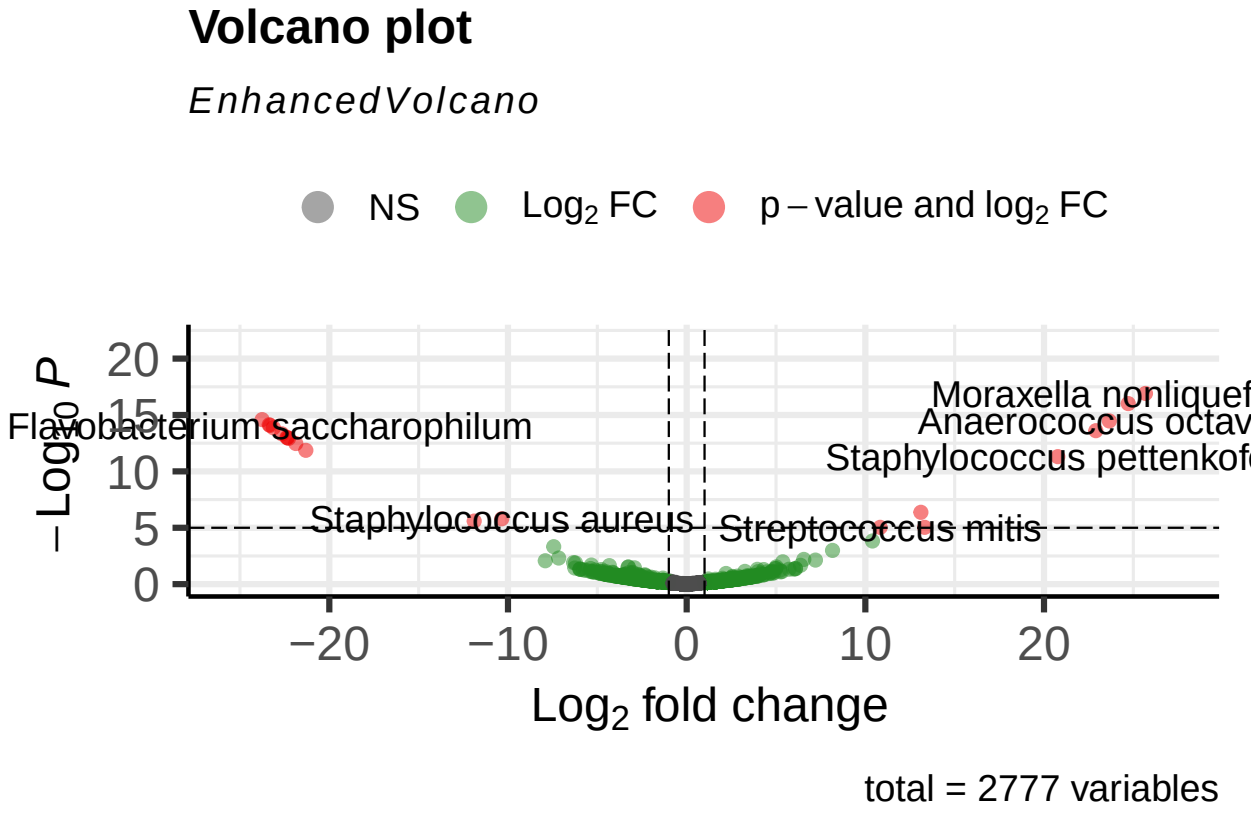
```
## converting counts to integer mode
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
deseq_ms_nasal = estimateSizeFactors(deseq_ms_nasal, type = 'poscounts')
deseq_ms_nasal = DESeq(deseq_ms_nasal, test="Wald", fitType="parametric")
```

```
## using pre-existing size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
deseq_ms_nasal_res = results(deseq_ms_nasal, cooksCutoff = FALSE)
deseq_ms_nasal_res <- cbind(deseq_ms_nasal_res,
                            as(tax_table(ps_ms_nasal)[rownames(deseq_ms_nasal_res), ], "matrix"))

EnhancedVolcano(deseq_ms_nasal_res,
  lab = deseq_ms_nasal_res$species,
  x = 'log2FoldChange',
  y = 'pvalue')
```



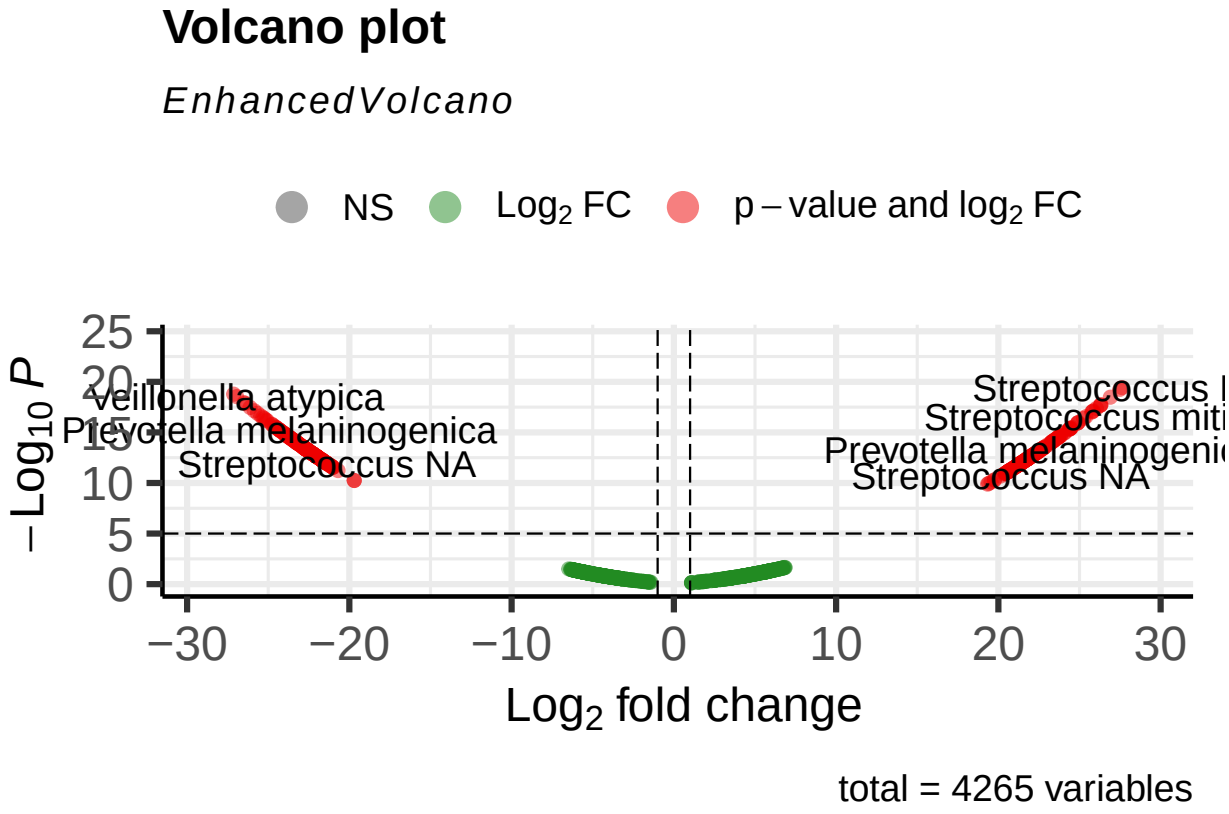
```
deseq_dada2_oral = phyloseq_to_deseq2(ps_dada2_oral, ~ status)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
deseq_dada2_oral = estimateSizeFactors(deseq_dada2_oral, type = 'poscounts')
deseq_dada2_oral = DESeq(deseq_dada2_oral, test="Wald", fitType="parametric")
```

```
## using pre-existing size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.

## final dispersion estimates

## fitting model and testing
```

```
deseq_dada2_oral_res = results(deseq_dada2_oral, cooksCutoff = FALSE)
deseq_dada2_oral_res <- cbind(deseq_dada2_oral_res,
                              as(tax_table(ps_dada2_oral)[rownames(deseq_dada2_oral_res), ], "matrix"))
EnhancedVolcano(deseq_dada2_oral_res,
  lab = paste0(deseq_dada2_oral_res$Genus, " ", deseq_dada2_oral_res$Species),
  x = 'log2FoldChange',
  y = 'pvalue')
```




```
deseq_dada2_nasal = phyloseq_to_deseq2(ps_dada2_nasal, ~ status)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
deseq_dada2_nasal = estimateSizeFactors(deseq_dada2_nasal, type = 'poscounts')
deseq_dada2_nasal = DESeq(deseq_dada2_nasal, test="Wald", fitType="parametric")
```

```
## using pre-existing size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.

## final dispersion estimates
```

```
## fitting model and testing
```

```
deseq_dada2_nasal_res = results(deseq_dada2_nasal, cooksCutoff = FALSE)
deseq_dada2_nasal_res <- cbind(deseq_dada2_nasal_res,
                               as(tax_table(ps_dada2_nasal)[rownames(deseq_dada2_nasal_res), ], "matrix"))
EnhancedVolcano(deseq_dada2_nasal_res,
  lab = paste0(deseq_dada2_nasal_res$Genus, " ", deseq_dada2_nasal_res$Species),
  x = 'log2FoldChange',
  y = 'pvalue')
```