botero_analysis

Sean Lu

2025-05-19

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
```

Load in processed Data

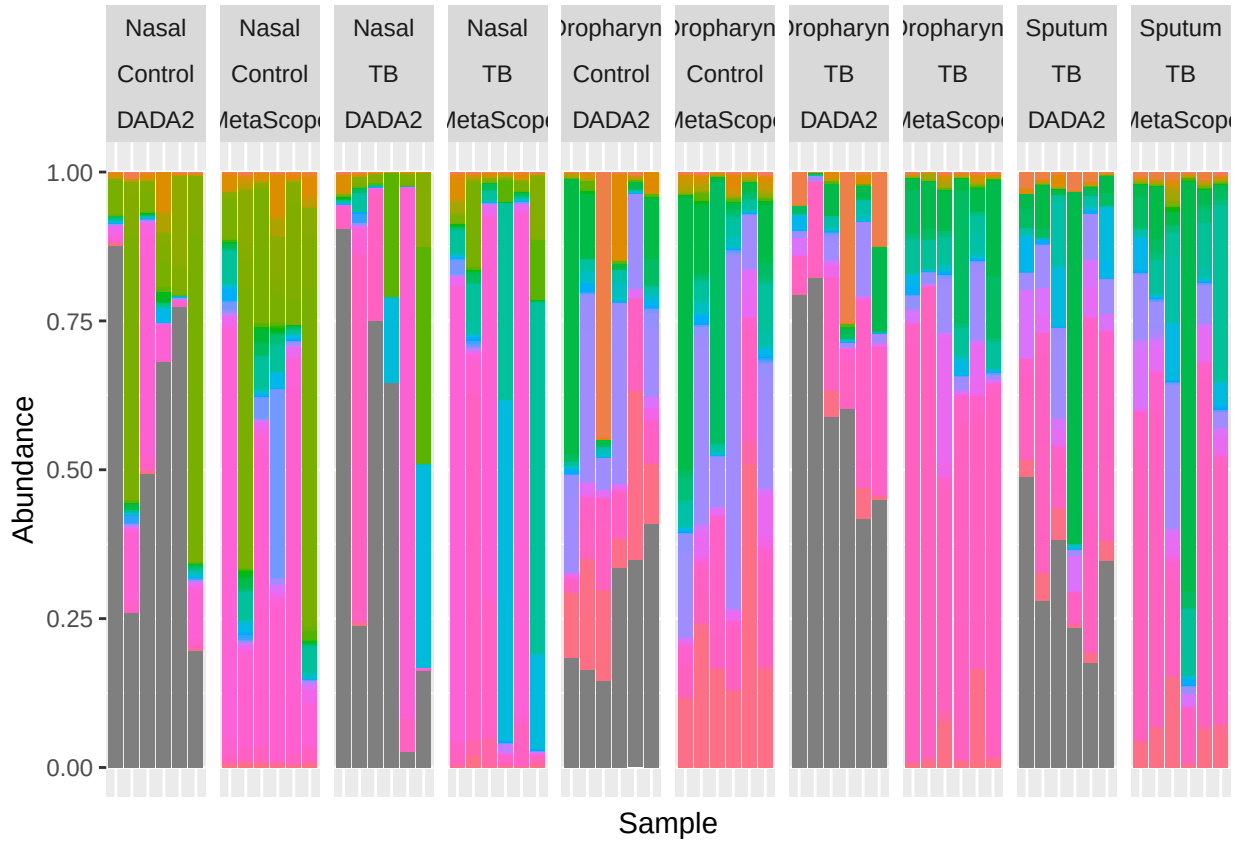
```
ps_dada2 <- readRDS("data_processed/botero_2014/ps_dada2.rds")
ps_ms <- readRDS("data_processed/botero_2014/ps_metascope_priors_trimmed.rds")

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")
```

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

Species Level Abundances

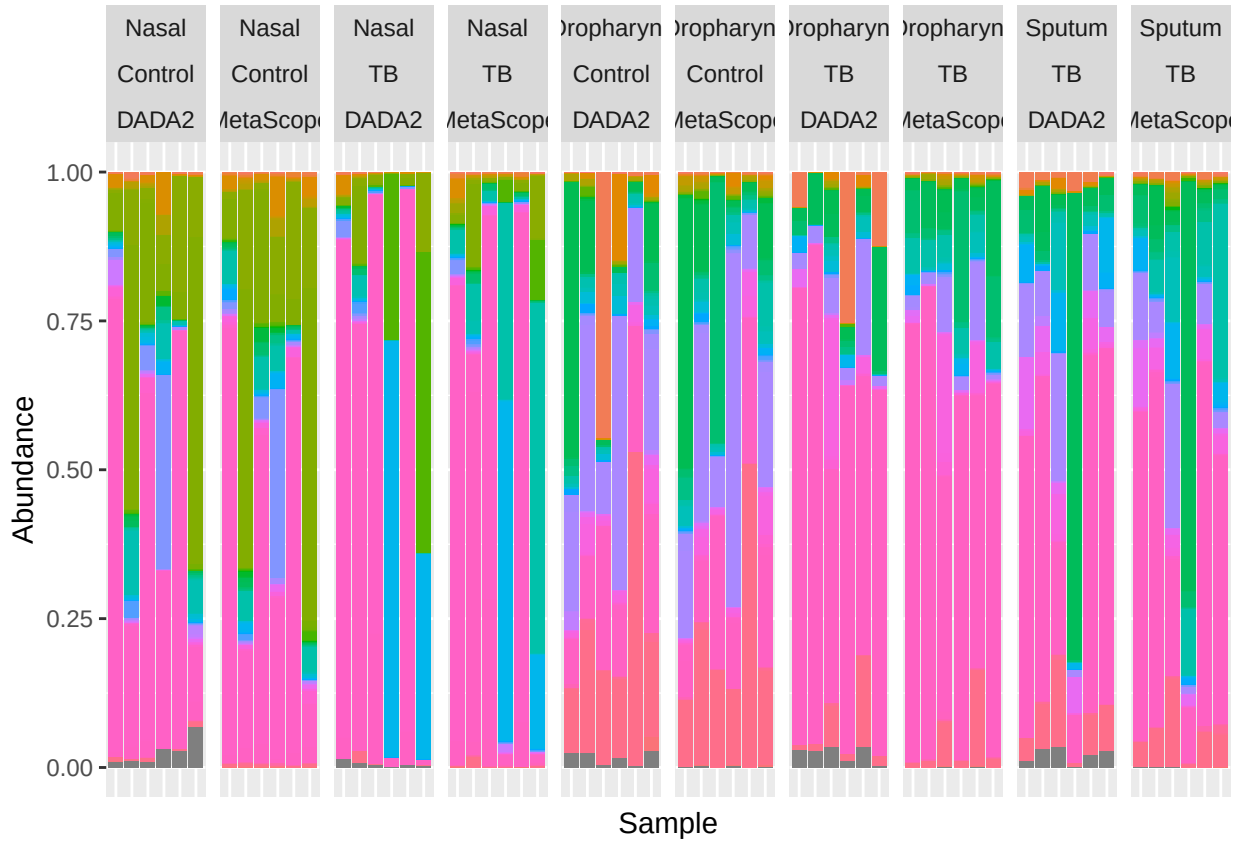
```
dada2_df <- psmelt(ps_dada2) |>
  dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species)))
dada2_df$pipeline = "DADA2"

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)

relab_species <- ggplot(merged_df, aes(x = Sample, y = Abundance, fill = Species)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = c(vars(Sample_type), vars(status), vars(pipeline)), scales = "free_x")
relab_species +
  theme(legend.position = "none",
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



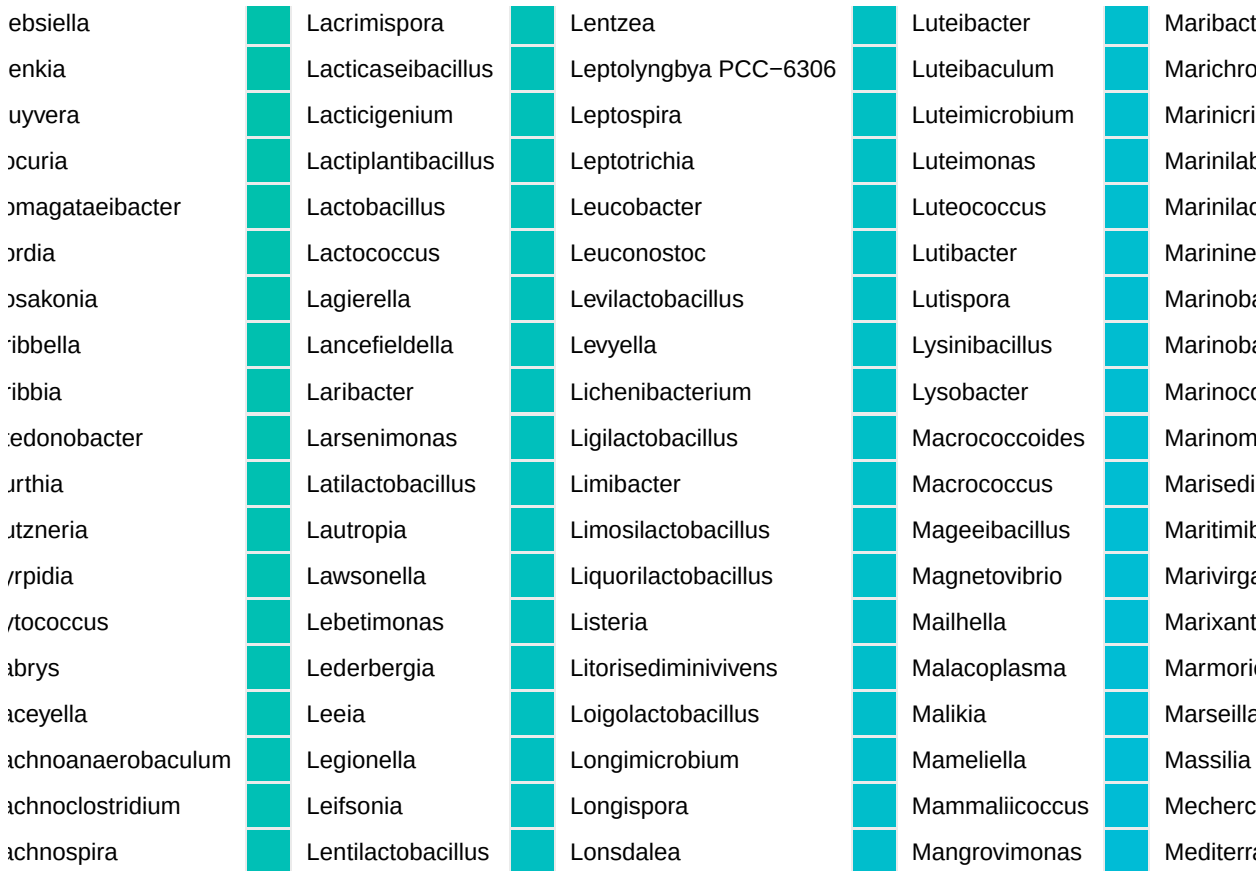

```
relab_species_legend <- get_legend(relab_species)
as_ggplot(relab_species_legend)
```

Genus Level Abundances

```
relab_genus <- ggplot(merged_df, aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = c(vars(Sample_type), vars(status), vars(pipeline)), scales = "free_x")
relab_genus +
  theme(legend.position = "none",
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

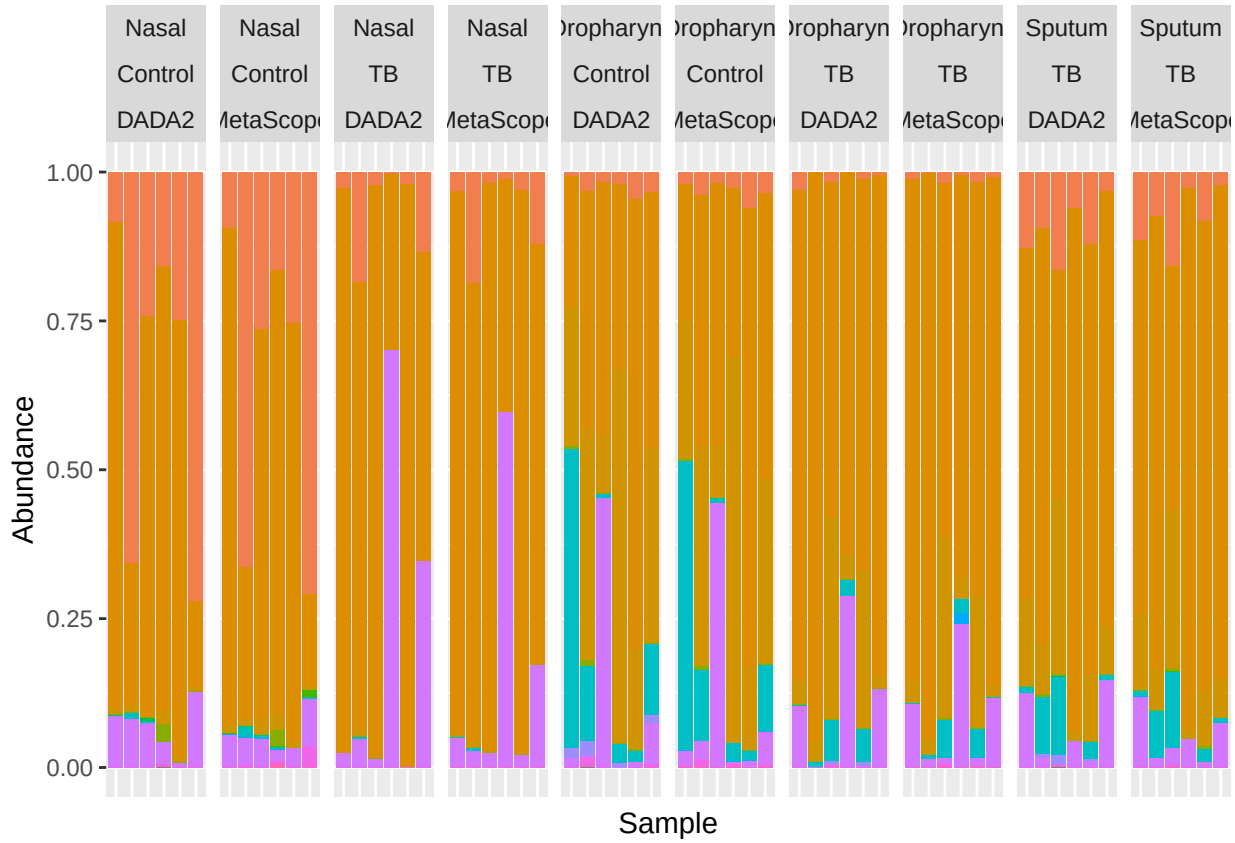



```
relab_genus_legend <- get_legend(relab_genus)
as_ggplot(relab_genus_legend)
```

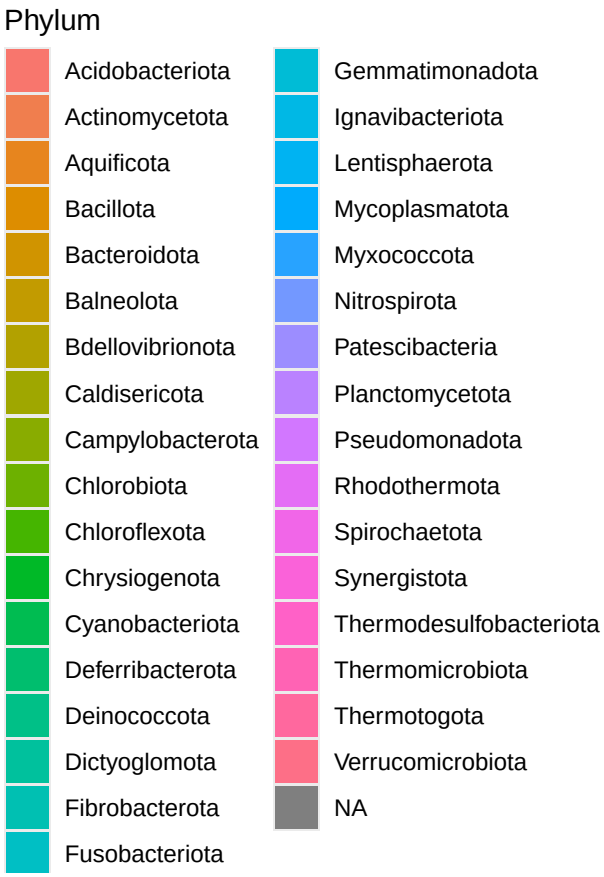


Phylum Level Abundances

```
relab_phylum <- ggplot(merged_df, aes(x = Sample, y = Abundance, fill = Phylum)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = c(vars(Sample_type), vars(status), vars(pipeline)), scales = "free_x")
relab_phylum +
  theme(legend.position = "none",
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

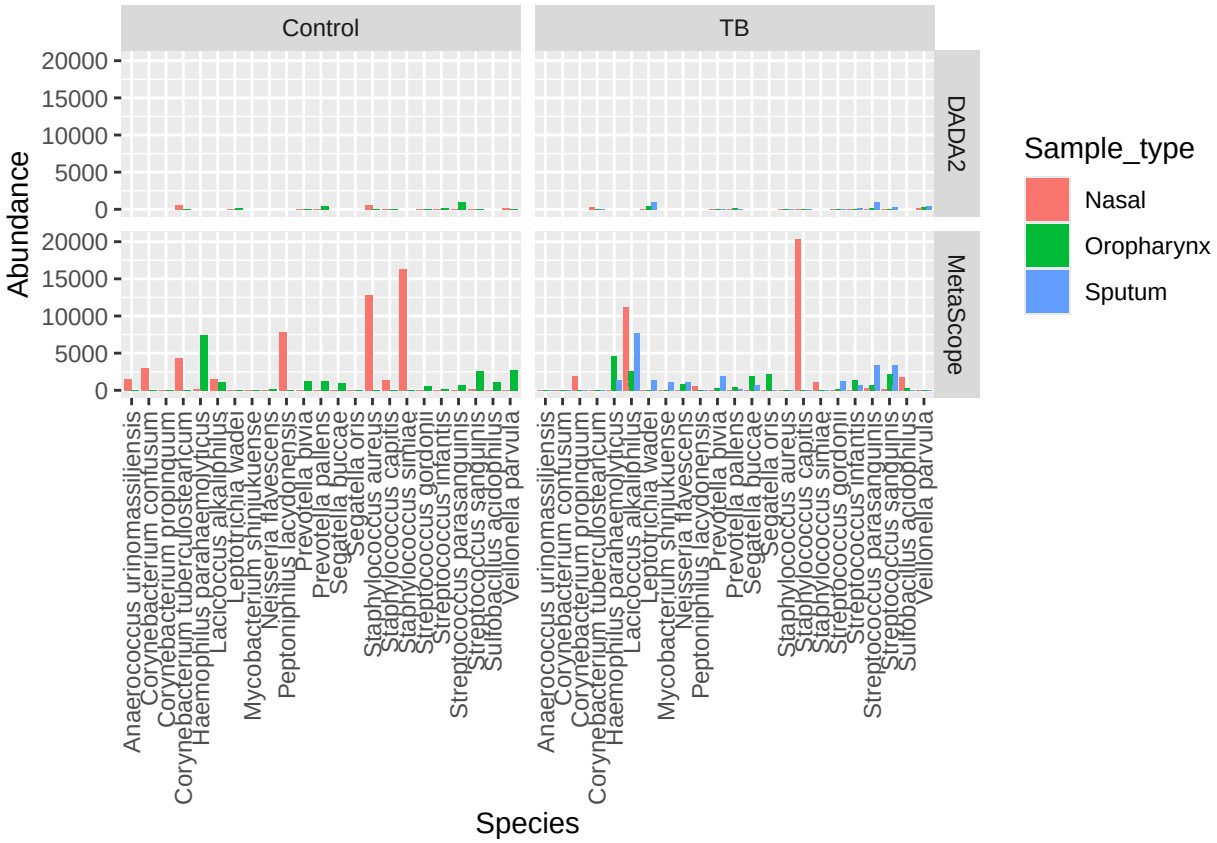



```
relab_phylum_legend <- get_legend(relab_phylum)
as_ggplot(relab_phylum_legend)
```

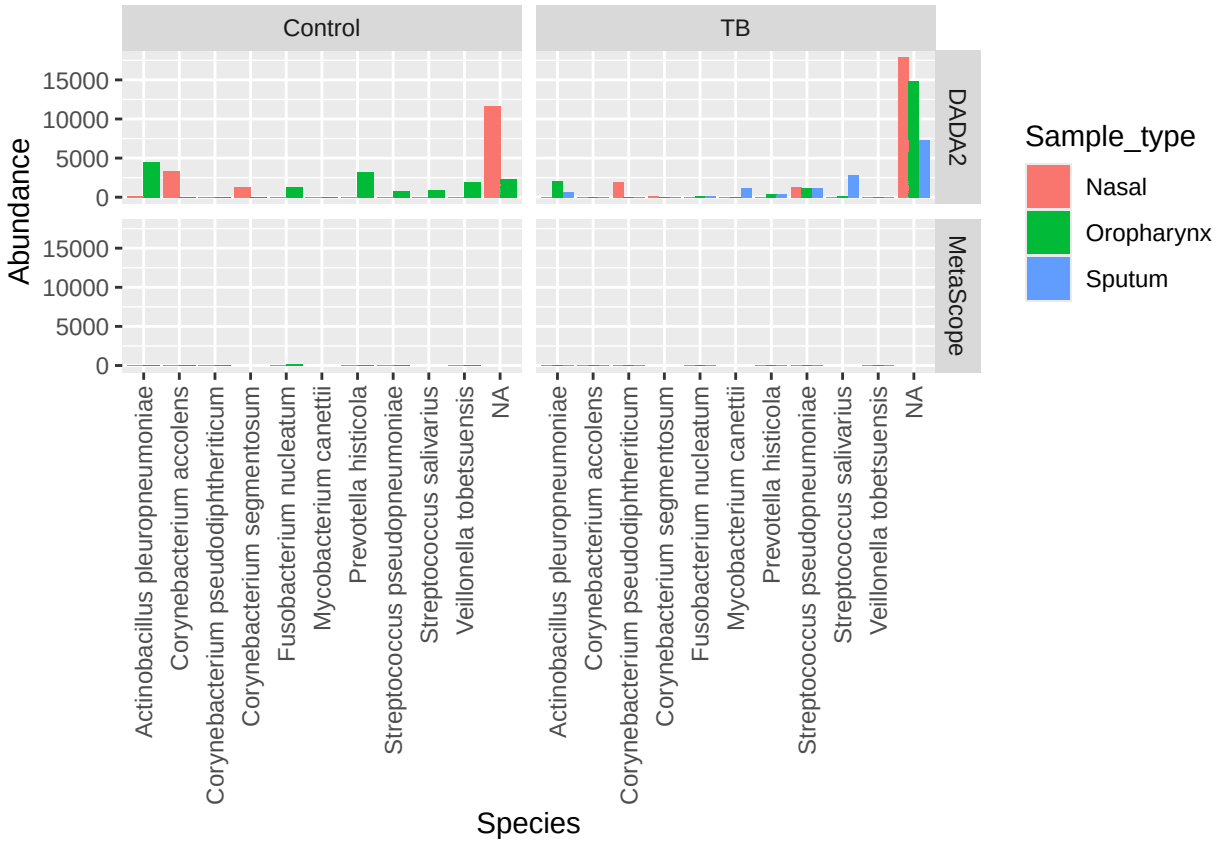


Filtered abundance barplots

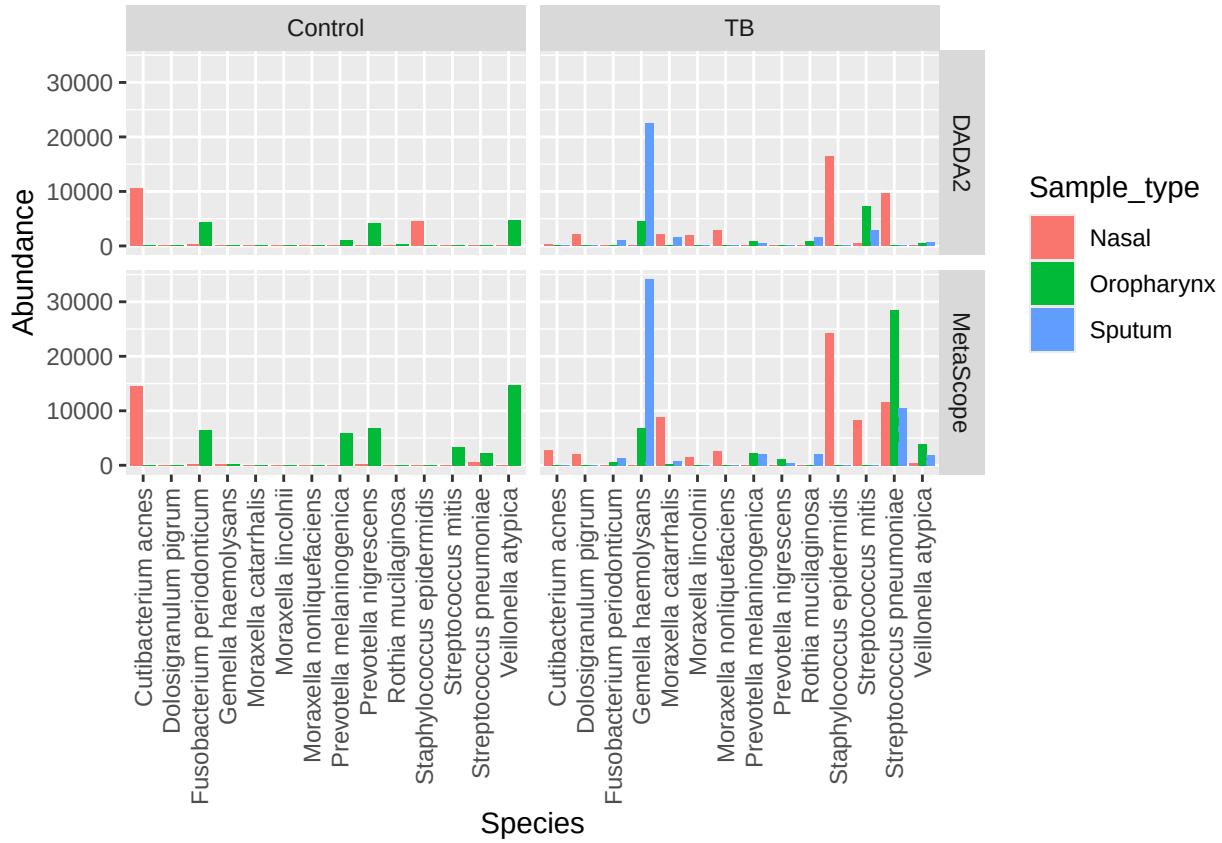
```
high_abund_dada2 <- merged_df |>
  dplyr::filter(Abundance > 1000) |>
  dplyr::filter(pipeline == "DADA2")

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
```

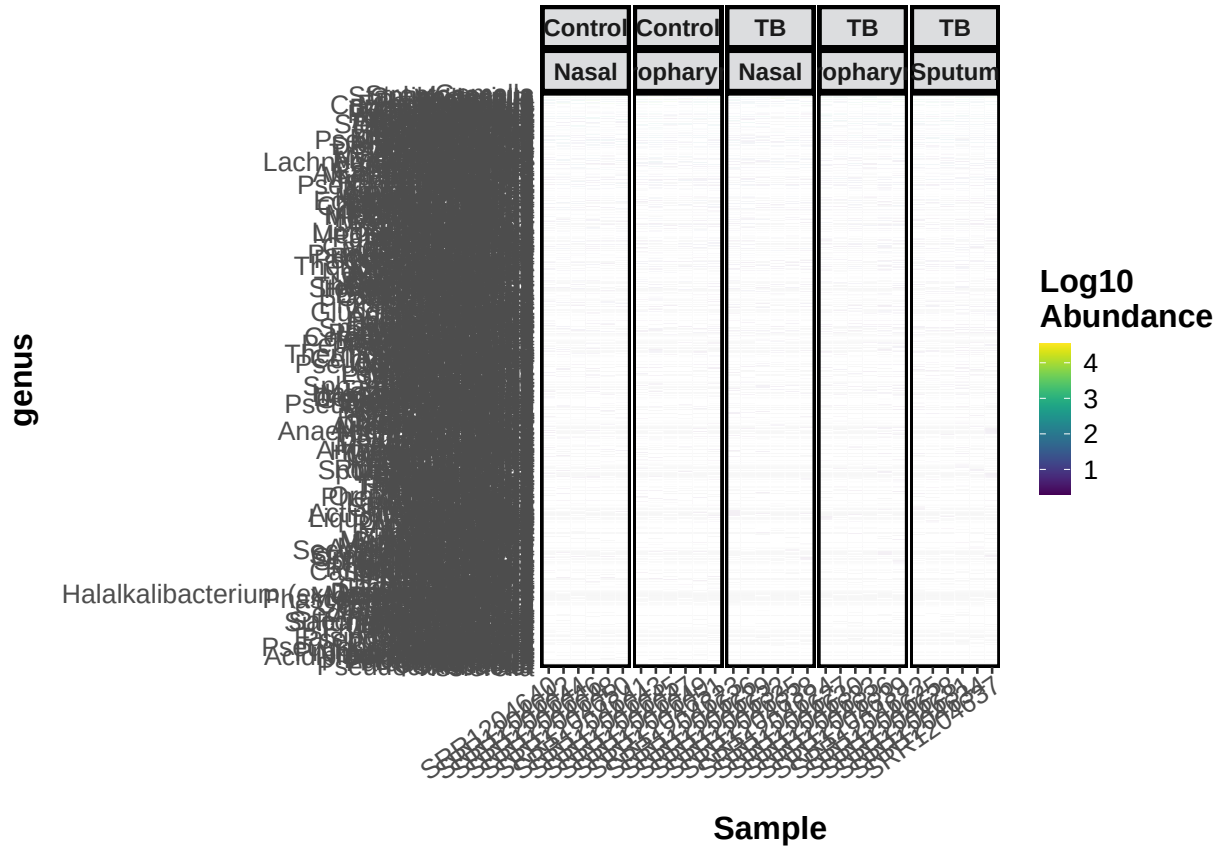



```
high_abund_species <- merged_df |>
  dplyr::filter(Species %in% select_species_both) |>
  ggplot(aes(fill=Sample_type, y=Abundance, x=Species)) +
  geom_bar(position="dodge", stat="identity") +
  facet_grid(vars(pipeline), vars(status)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5,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

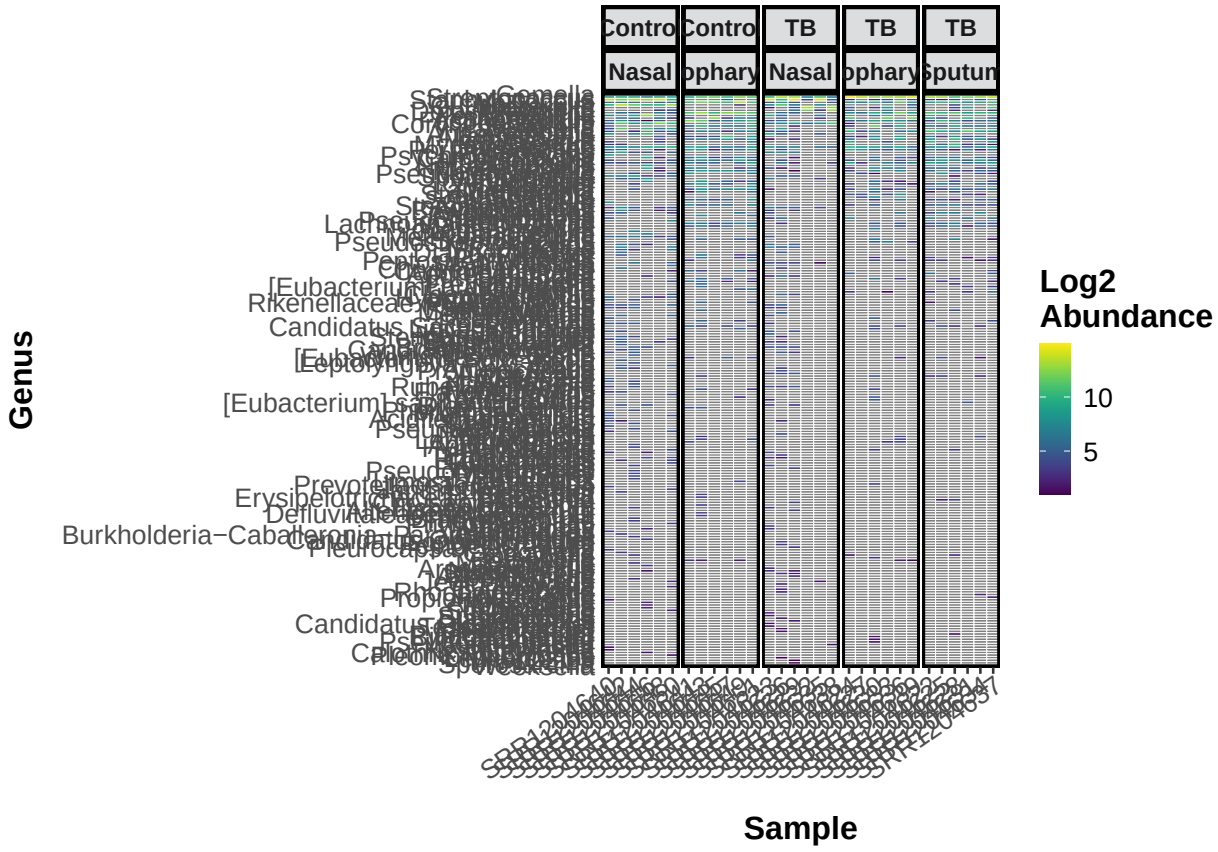


Heatmaps

```
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"))) +
  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"))) +
  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"))) +
  geom_boxplot()
```

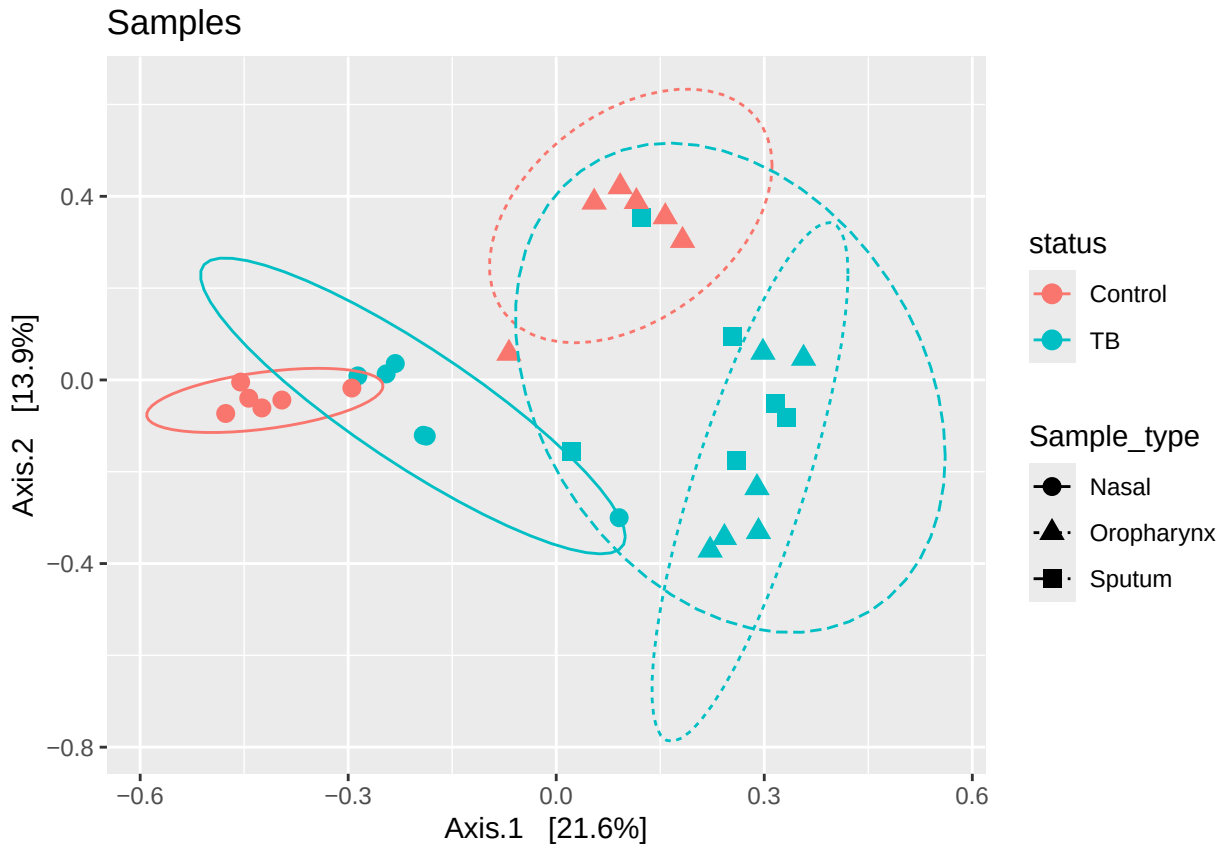
```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided c
## any singletons. This is highly suspicious. Results of richness
```

```
## 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_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"))) +
  geom_boxplot()
```

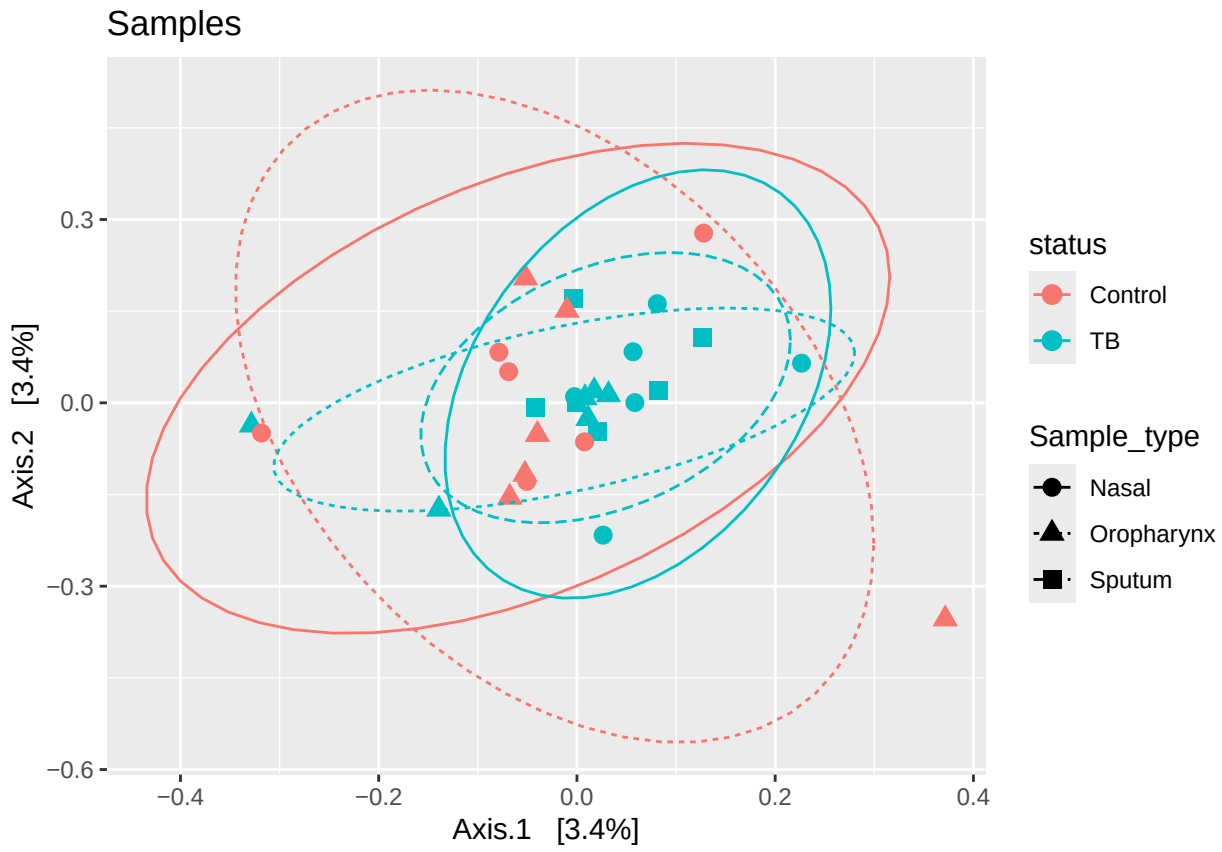
PCOA Plots

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

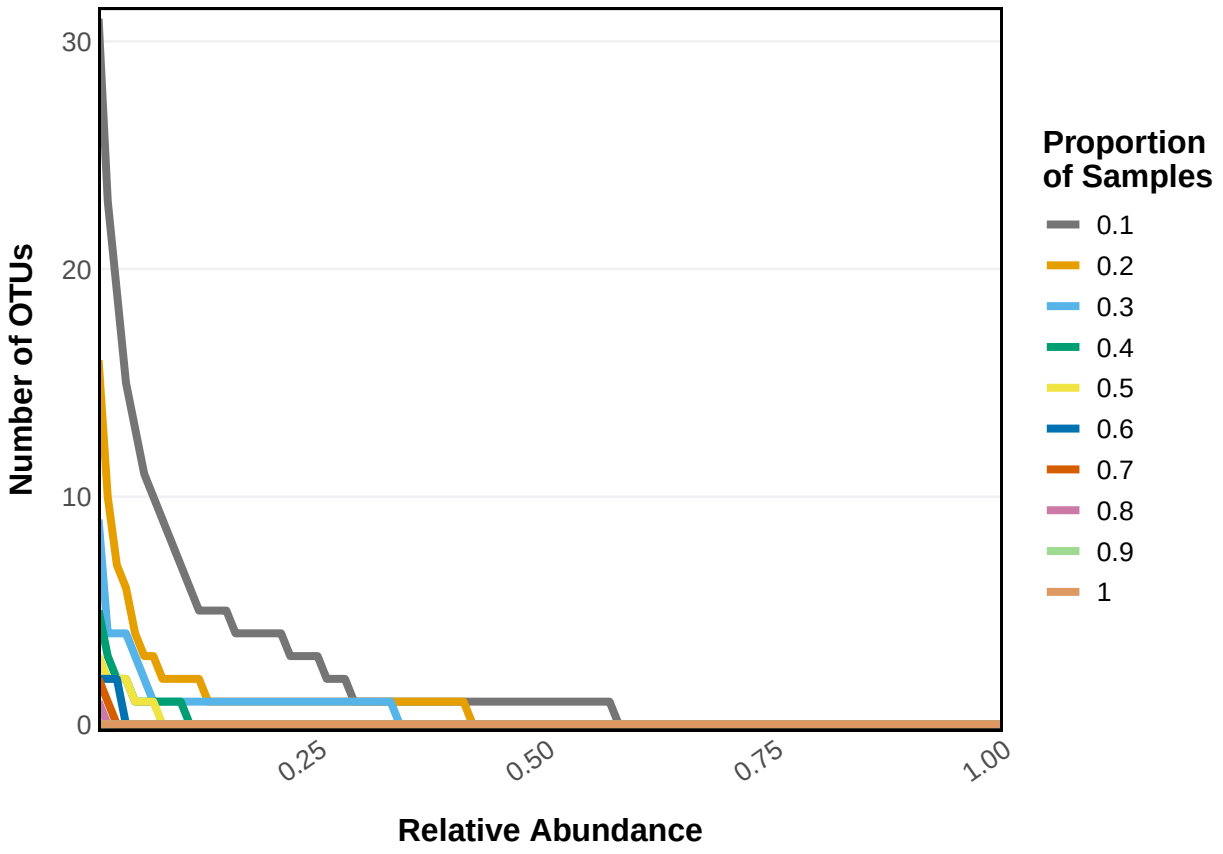



```
ps_dada2.ord <- ordinate(ps_dada2, "PCoA", "bray")
plot_ordination(ps_dada2, ps_dada2.ord, type="samples", color="status", shape="Sample_type", title="Samp
  geom_point(size=3) +
  stat_ellipse(
    aes(linetype=Sample_type))
```

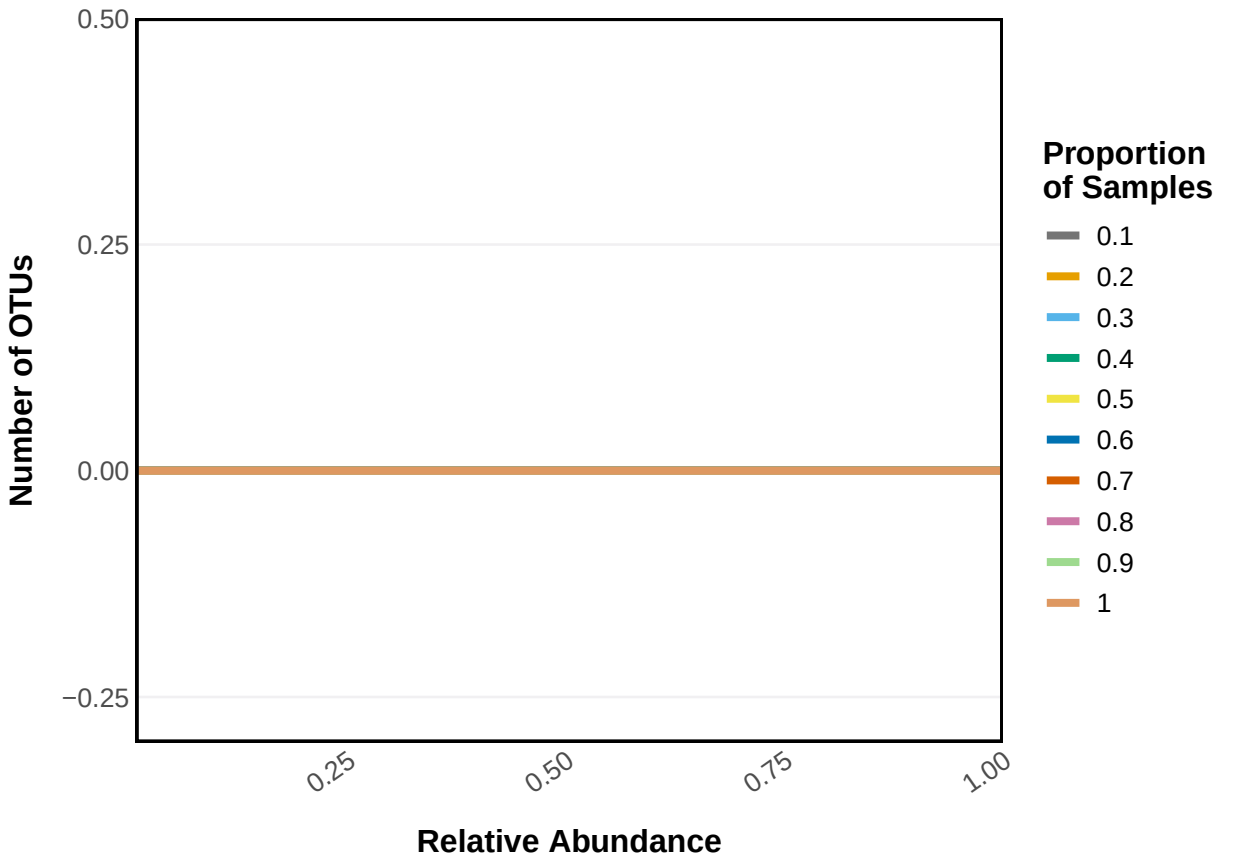
```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```




```
taxa_core_graph(ps_ms, treatment = NULL, subset = NULL,
  frequencies = seq(0.1, 1, 0.1), abundance_thresholds = seq(0.01, 1, 0.01),
  colors = 'default')
```




```
taxa_core_graph(ps_dada2, treatment = NULL, subset = NULL,
  frequencies = seq(0.1, 1, 0.1), abundance_thresholds = seq(0.01, 1, 0.01),
  colors = 'default')
```




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

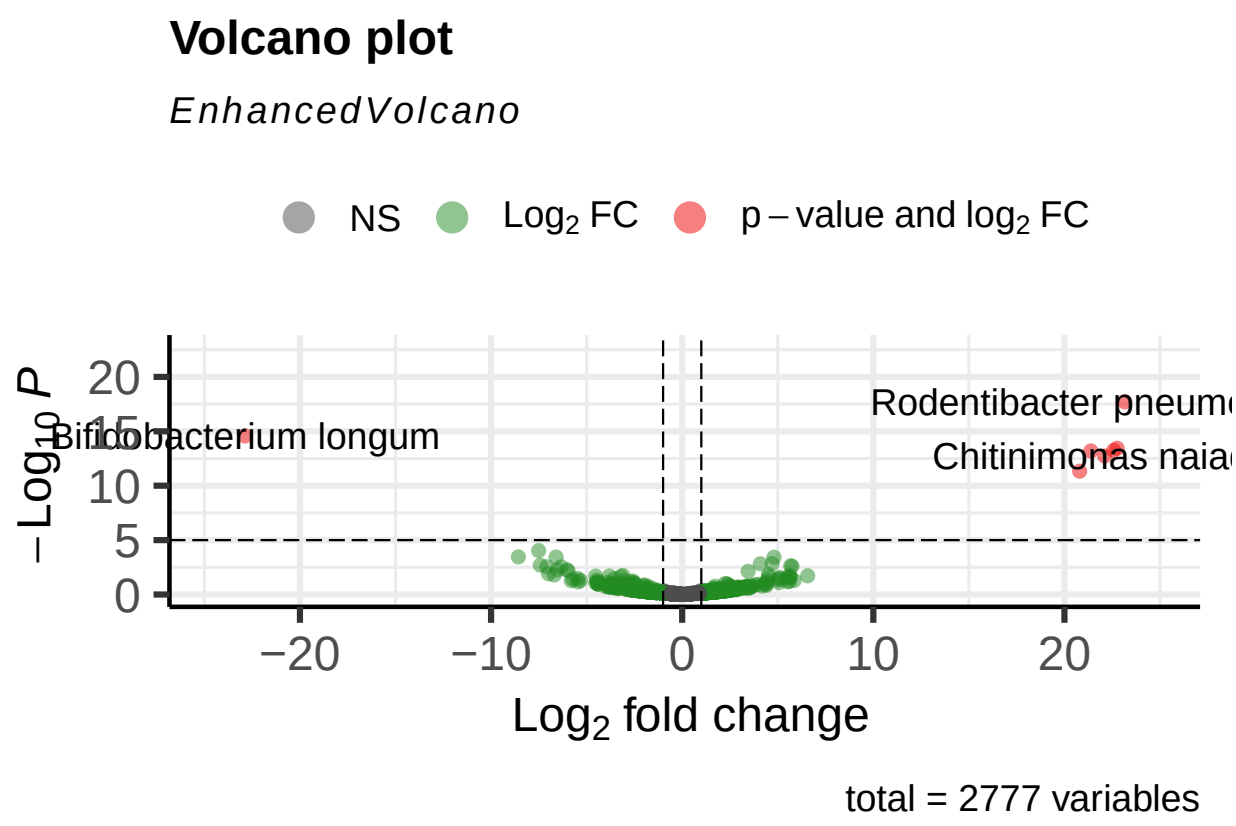
```
## converting counts to integer mode
```

```
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')
```



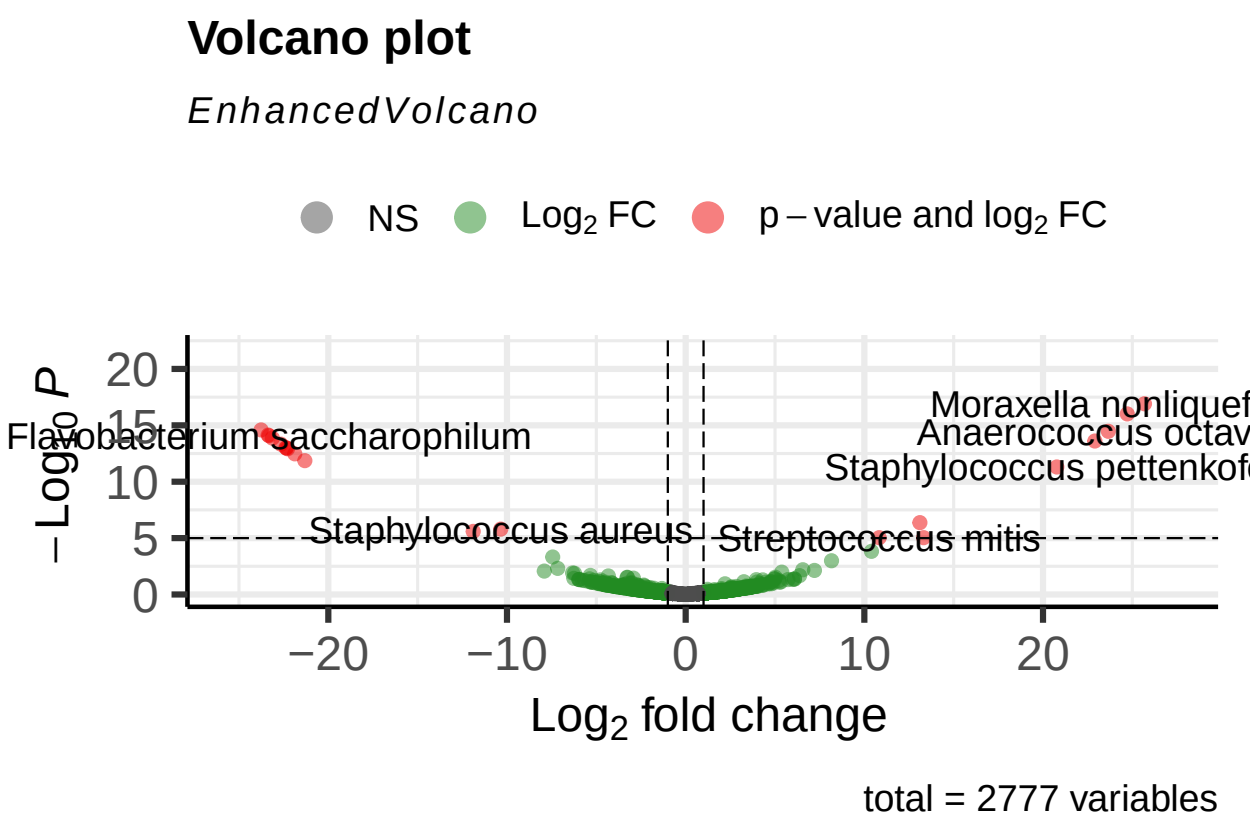

```
deseq_ms_nasal = phyloseq_to_deseq2(ps_ms_nasal, ~ status)
```

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

```
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)
```

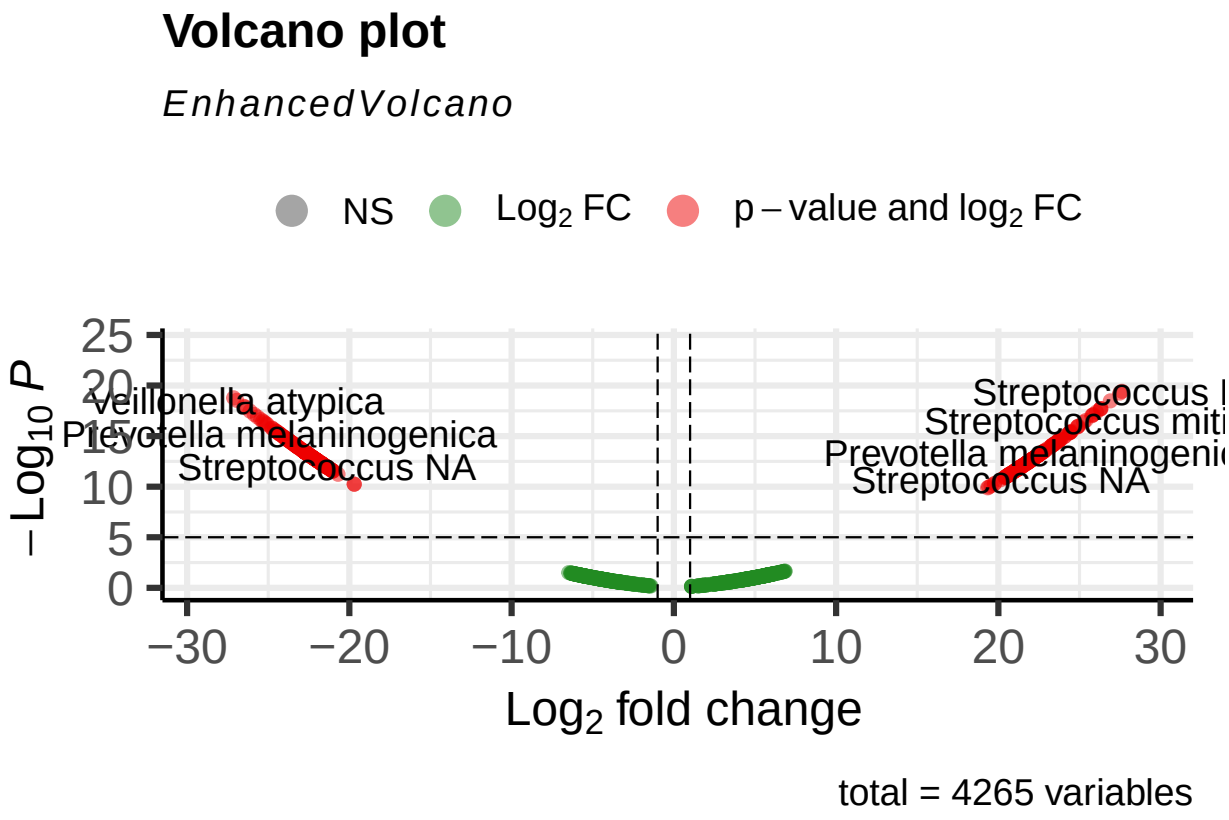
```
## converting counts to integer mode
```

```
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
```

```
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')
```