botero_analysis

Sean Lu

2025-09-05

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
```

```
library("tidyr")
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:ggVennDiagram':
##
##     unite

## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
library("rstatix")
```

```
##
## Attaching package: 'rstatix'

## The following object is masked from 'package:IRanges':
##
##     desc

## The following object is masked from 'package:stats':
##
##     filter
```

Load in processed Data

```
ps_dada2 <- readRDS("data_processed/botero_2014/ps_dada2.rds")
ps_qiime2 <- readRDS("data_processed/botero_2014/ps_qiime2.rds")
ps_ms <- readRDS("data_processed/botero_2014/ps_metascope.rds")

ps_dada2_oral <- subset_samples(ps_dada2, Sample_type == "Oropharynx")
ps_qiime2_oral <- subset_samples(ps_qiime2, Sample_type == "Oropharynx")
ps_ms_oral <- subset_samples(ps_ms, Sample_type == "Oropharynx")
ps_dada2_nasal <- subset_samples(ps_dada2, Sample_type == "Nasal")
ps_qiime2_nasal <- subset_samples(ps_qiime2, Sample_type == "Nasal")
ps_ms_nasal <- subset_samples(ps_ms, Sample_type == "Nasal")

# Use ps_melt to generate tidy format dataframes
dada2_df <- psmelt(ps_dada2) |>
  dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species))) |>
  dplyr::rename(Superkingdom = Kingdom)
dada2_df$pipeline = "DADA2"
```

```r
qiime2_df <- psmelt(ps_qiime2) |>
  dplyr::select(-Confidence)
qiime2_df$pipeline = "QIIME2"

ms_df <- psmelt(ps_ms)
ms_df$pipeline = "MetaScope"

# Merge all data together
merged_df <- rbind(dada2_df, qiime2_df, ms_df)
merged_df$pipeline <- factor(merged_df$pipeline,
                             levels = c("MetaScope", "DADA2", "QIIME2"))

# Generate relative abundance psmelt dataframes
dada2_relab_df <- phylosmith::relative_abundance(ps_dada2) |>
  phyloseq::psmelt() |>
  dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species))) |>
  dplyr::rename(Superkingdom = Kingdom)
dada2_relab_df$pipeline = "DADA2"

qiime2_relab_df <- phylosmith::relative_abundance(ps_qiime2) |>
  phyloseq::psmelt() |>
  dplyr::select(-Confidence)
qiime2_relab_df$pipeline = "QIIME2"
qiime2_relab_df <- qiime2_relab_df |>
  dplyr::mutate(Species = gsub("_", " ", Species))

ms_relab_df <- phylosmith::relative_abundance(ps_ms) |>
  phyloseq::psmelt()
ms_relab_df$pipeline = "MetaScope"
colnames(ms_relab_df) <- c("OTU", "Sample", "Abundance", "Sequencing_Type", "Patient",
                     "Sample_type", "status", "Superkingdom", "Phylum", "Class",
                     "Order", "Family", "Genus", "Species", "pipeline")

merged_relab_df <- rbind(dada2_relab_df, qiime2_relab_df, ms_relab_df)
```

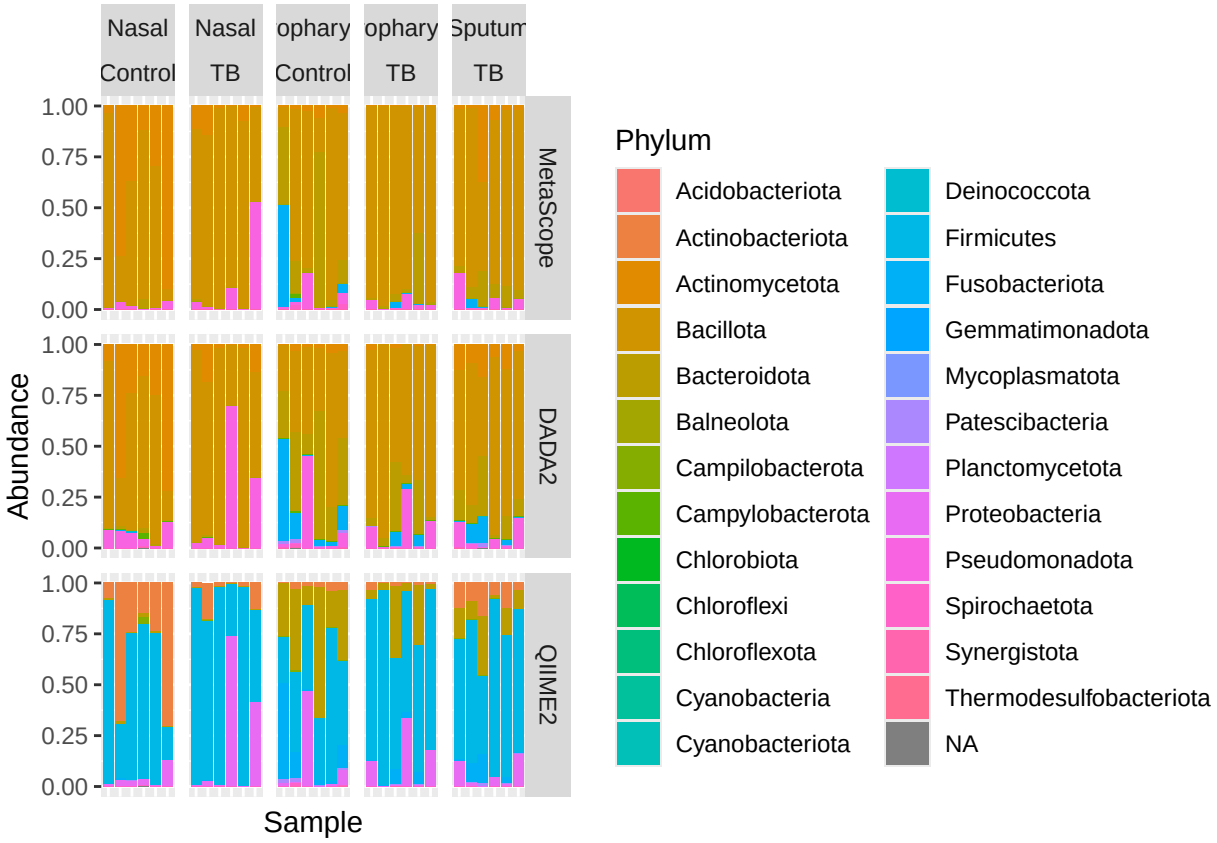
The DADA2 results are generated from the `dada2_botero.Rmd` file. The QIIME2 results are generated from the `qiime2_botero.sh` script. The MetaScope results are generated from the `process_metascope_id.R` scripts.

Plotting relative abundances of MetaScope and DADA2

Phylum Level Abundances

```r
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()
```

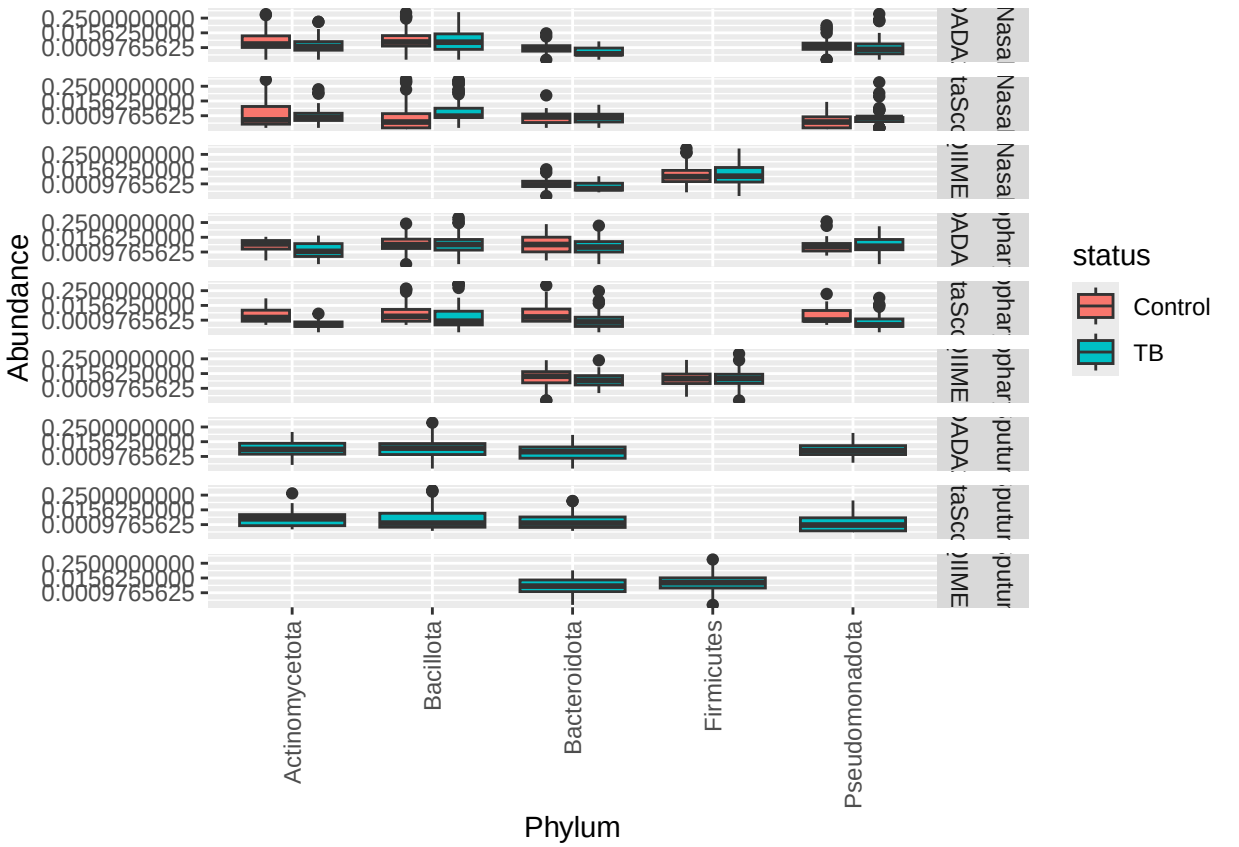
```
relab_phylum <- ggplot(merged_df,
                       aes(x = Sample, y = Abundance, fill = Phylum)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status),
             rows = vars(pipeline), scales = "free_x") +
  scale_fill_discrete() +
  theme(axis.text.x=element_blank(),
        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 110990 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

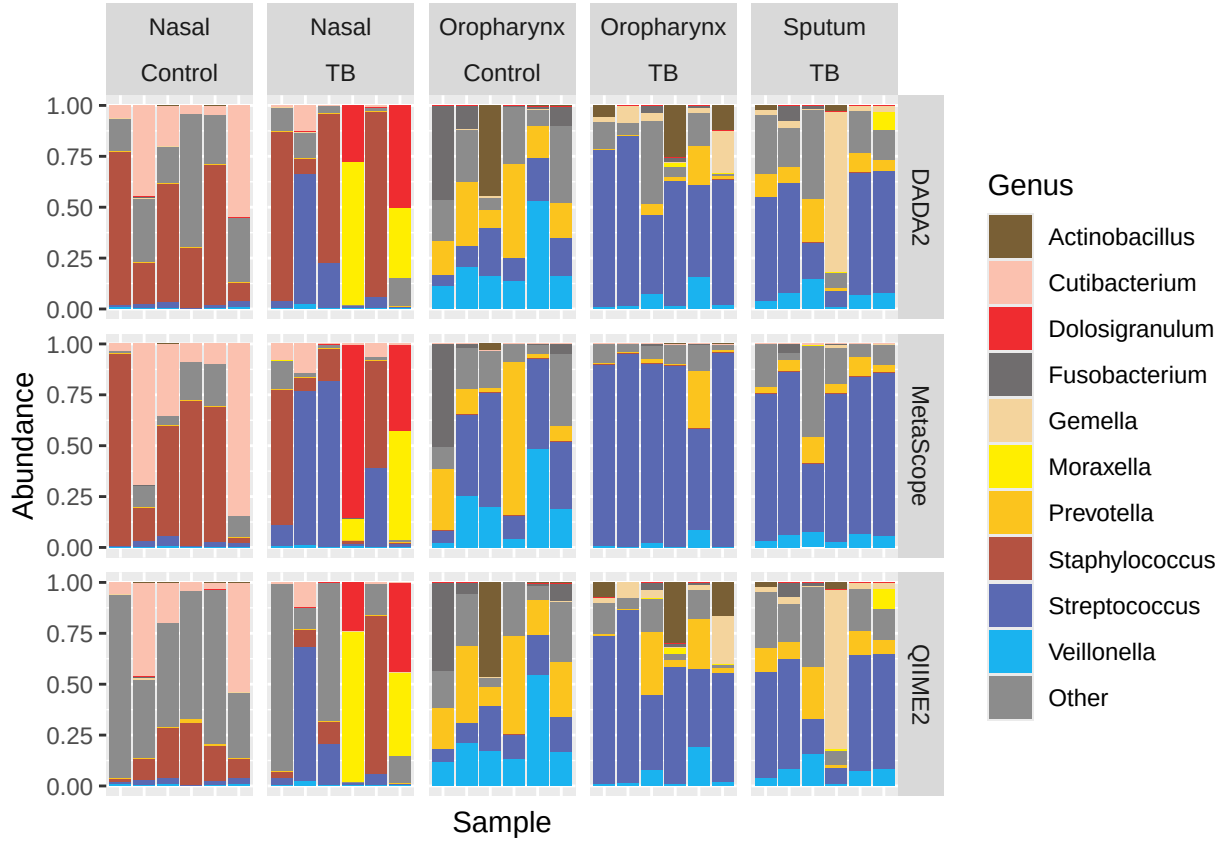


At Phylum level taxonomies, both DADA2 and MetaScope show similar results. Nasal samples in the controls show increased abundance of Acidobacteriota compared to TB sample and a decreased relative abundance in Pseudomonadota in controls relative to TB samples. The oropharynx samples mild decrease in Baciollota 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 = 10) |>
  pull(Genus) |>
  sort()

genera_df <- merged_relab_df |>
  mutate(Genus = ifelse(Genus %in% top_genera, Genus, "Other"))

relab_genus <- ggplot(genera_df,
                      aes(x = Sample, y = Abundance, fill = Genus)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status),
             rows = vars(pipeline),
             scales = "free_x") +
```

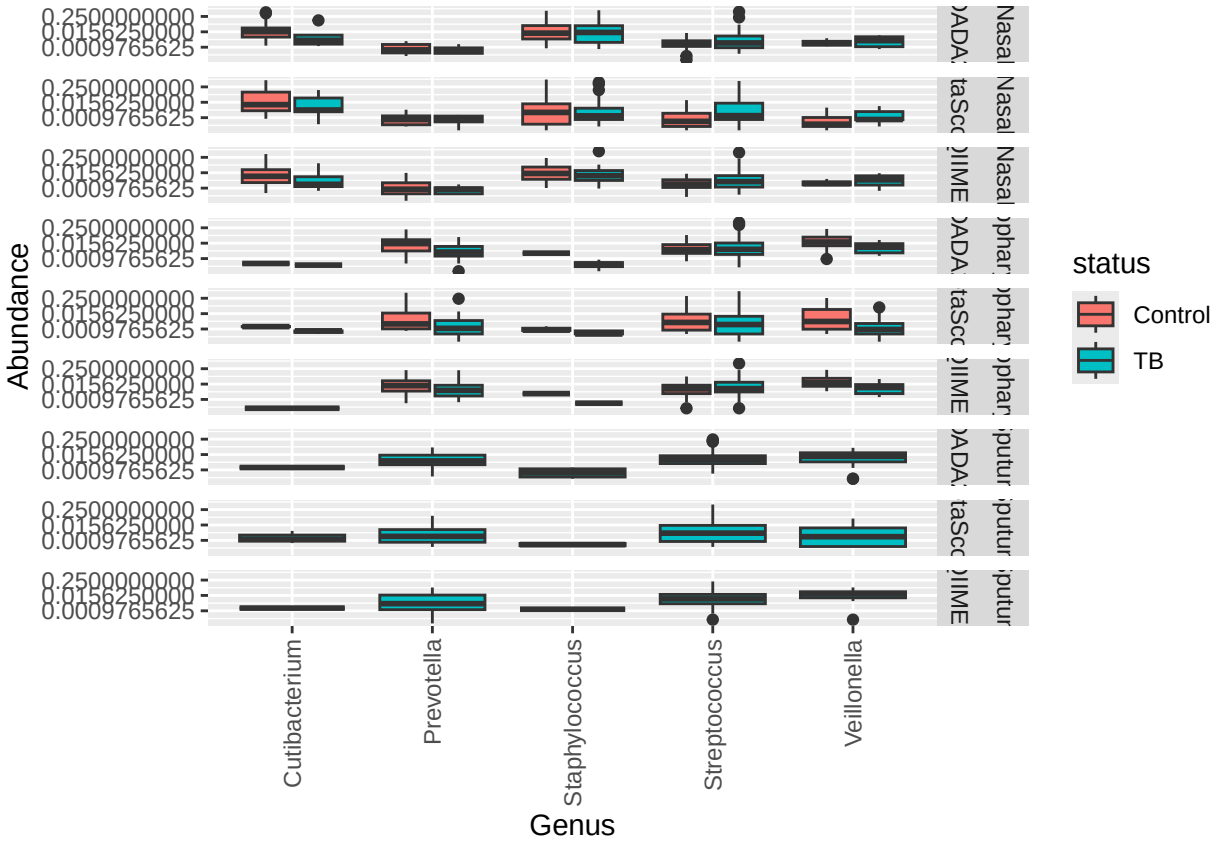
```
  scale_fill_manual(values = c(paletteer::paletteer_d("khroma::soil", 10),"gray55"),
                    breaks = c(top_genera, "Other")) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
relab_genus
```




```
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 49908 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

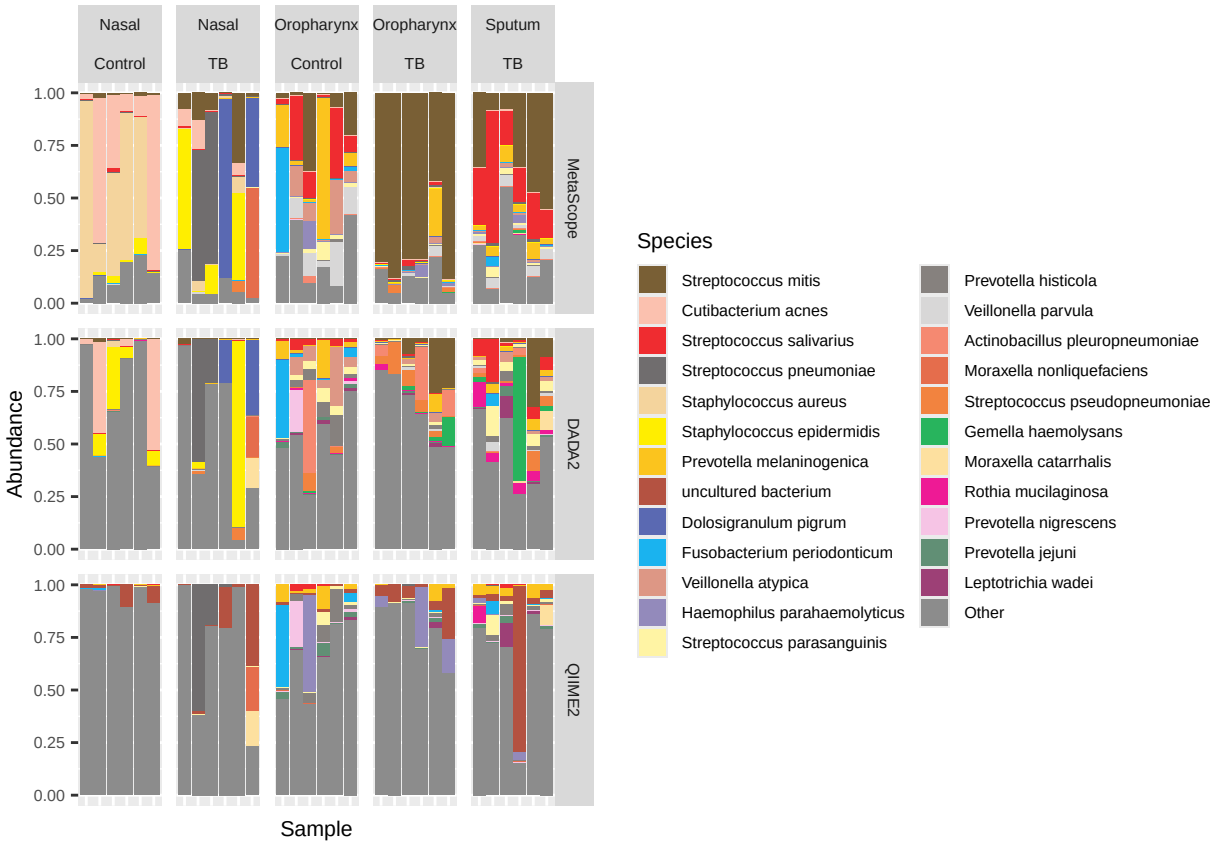


At the genus level, still DADA2 and MetaScope have similar relative abundances and identify the same genera that are differentially expressed. Notably, DADA2 identifies more NAs

Species Level Abundances

```
top_species <- merged_relab_df |>
  dplyr::group_by(Species) |>
  dplyr::summarise(total_abund = sum(Abundance), .groups = "drop") |>
  dplyr::filter(!is.na(Species)) |>
  #dplyr::filter(Species != "uncultured bacterium") |>
  slice_max(order_by=total_abund, n = 24) |>
  pull(Species)

num_colors <- 24
wheel_colors <- c(paletteer::paletteer_d("khroma::soil", num_colors),"gray55")
merged_df_plot <- merged_relab_df |>
  mutate(Species = ifelse(Species %in% top_species, Species, "Other"))
merged_df_plot$Species <- factor(merged_df_plot$Species, levels = c(top_species, "Other"))
merged_df_plot$pipeline <- factor(merged_df_plot$pipeline, levels = c("MetaScope", "DADA2", "QIIME2"))
```

```
color_map <- setNames(wheel_colors, levels(merged_relab_df$Species))

relab_species <- ggplot(merged_df_plot,
                        aes(x = Sample, y = Abundance, fill = Species)) +
  geom_bar(position ="fill",stat = "identity") +
  facet_grid(cols = vars(Sample_type, status),
             rows = vars(pipeline),
             scales = "free_x") +
  scale_fill_manual(values = color_map,
                    breaks = levels(merged_df_plot$Species)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        text=element_text(size=8),
        legend.key.size=unit(4,"mm"))
relab_species
```

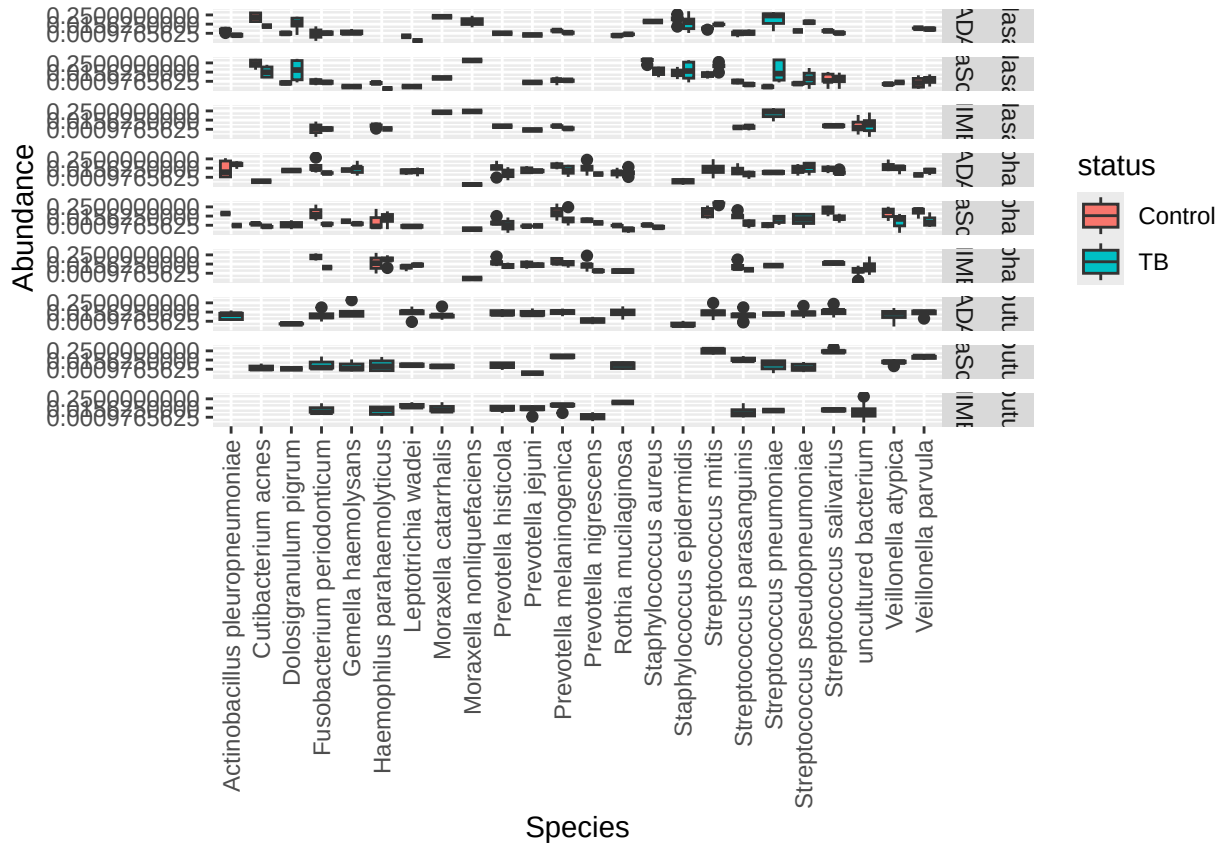



```
ggsave(filename="figures/p7_botero_relab.png", plot=relab_species, dpi=450,
       width=180,height=90,units="mm",device="png")


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 21454 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
merged_df |> dplyr::filter(pipeline == "MetaScope", Sample_type == "Nasal") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## ‘summarise()‘ has grouped output by ’Species’, ’Genus’. You can override using
## the ‘.groups‘ argument.
```

```
## # A tibble: 1,192 x 4
## # Groups:   Species, Genus [596]
##    Species                   Genus           status   mean
##    <chr>                     <chr>           <chr>   <dbl>
##  1 Staphylococcus aureus     Staphylococcus  Control 2889
##  2 Cutibacterium acnes       Cutibacterium   Control 2444.
```

```
##  3 Streptococcus pneumoniae   Streptococcus   TB      1261.
##  4 Staphylococcus epidermidis Staphylococcus  TB       349.
##  5 Streptococcus mitis        Streptococcus   TB       317
##  6 Cutibacterium acnes        Cutibacterium   TB       202.
##  7 Corynebacterium accolens   Corynebacterium Control  156.
##  8 Dolosigranulum pigrum      Dolosigranulum  TB       152.
##  9 Moraxella nonliquefaciens  Moraxella       TB       132.
## 10 Staphylococcus epidermidis Staphylococcus  Control  106.
## # i 1,182 more rows
```

```
merged_df |> dplyr::filter(pipeline == "DADA2", Sample_type == "Nasal") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 910 x 4
## # Groups:   Species, Genus [455]
##    Species                    Genus             status   mean
##    <chr>                      <chr>             <chr>   <dbl>
##  1 Corynebacterium accolens   Corynebacterium   Control  561.
##  2 Cutibacterium acnes        Cutibacterium     Control  508.
##  3 Streptococcus pneumoniae   Streptococcus     TB       448.
##  4 <NA>                       Staphylococcus    TB       208.
##  5 Staphylococcus epidermidis Staphylococcus    TB       192.
##  6 <NA>                       Staphylococcus    Control  185.
##  7 Moraxella nonliquefaciens  Moraxella         TB       165.
##  8 <NA>                       Peptoniphilus     Control  142.
##  9 <NA>                       Dolosigranulum    TB       125.
## 10 <NA>                       Psychroglaciecola Control  115.
## # i 900 more rows
```

```
merged_df |> dplyr::filter(pipeline == "QIIME2", Sample_type == "Nasal") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 678 x 4
## # Groups:   Species, Genus [339]
##    Species                   Genus          status   mean
##    <chr>                     <chr>          <chr>   <dbl>
##  1 Streptococcus_pneumoniae  Streptococcus  TB      468.
##  2 Moraxella_nonliquefaciens Moraxella      TB      187.
##  3 <NA>                      Cutibacterium  Control 129.
##  4 <NA>                      Peptoniphilus  Control 105.
##  5 Campylobacter_ureolyticus Campylobacter  Control  99.3
##  6 <NA>                      Lawsonella     Control  96.3
```

```
##  7 <NA>                      Moraxella      TB       91.2
##  8 <NA>                      Staphylococcus TB       79.6
##  9 Lawsonella_clevelandensis Lawsonella     TB       77.2
## 10 Moraxella_catarrhalis     Moraxella      TB       72.2
## # i 668 more rows
```

```
merged_df |> dplyr::filter(pipeline == "MetaScope", Sample_type == "Oropharynx") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 1,192 x 4
## # Groups:   Species, Genus [596]
##    Species                         Genus         status    mean
##    <chr>                           <chr>         <chr>    <dbl>
##  1 Streptococcus mitis             Streptococcus TB      3634.
##  2 Streptococcus salivarius        Streptococcus Control  266
##  3 Prevotella melaninogenica       Prevotella    Control  206.
##  4 Veillonella atypica             Veillonella   Control  177.
##  5 Streptococcus mitis             Streptococcus Control  169.
##  6 Prevotella melaninogenica       Prevotella    TB       138.
##  7 Fusobacterium periodonticum     Fusobacterium Control  119.
##  8 Streptococcus chosunense        Streptococcus TB       114.
##  9 Veillonella parvula             Veillonella   Control   82.8
## 10 Streptococcus pseudopneumoniae  Streptococcus TB        71.2
## # i 1,182 more rows
```

```
merged_df |> dplyr::filter(pipeline == "DADA2", Sample_type == "Oropharynx") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 910 x 4
## # Groups:   Species, Genus [455]
##    Species                          Genus          status   mean
##    <chr>                            <chr>          <chr>   <dbl>
##  1 Streptococcus vestibularis       Streptococcus  Control 159.
##  2 Prevotella nigrescens            Prevotella     Control 111.
##  3 Veillonella tobetsuensis         Veillonella    Control 108.
##  4 Veillonella atypica              Veillonella    Control  72.6
##  5 Streptococcus mitis              Streptococcus  TB       69.3
##  6 <NA>                             Streptococcus  TB       59.3
##  7 Actinobacillus pleuropneumoniae  Actinobacillus TB       59.0
##  8 Fusobacterium periodonticum      Fusobacterium  Control  58.7
##  9 <NA>                             Xylanibacter   Control  53.2
## 10 Gemella haemolysans              Gemella        TB       51
## # i 900 more rows
```

```
merged_df |> dplyr::filter(pipeline == "QIIME2", Sample_type == "Oropharynx") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 678 x 4
## # Groups:   Species, Genus [339]
##    Species                      Genus          status   mean
##    <chr>                        <chr>          <chr>   <dbl>
##  1 Leptotrichia-like_sp.        uncultured     Control 209
##  2 Prevotella_nigrescens        Prevotella     Control  77.7
##  3 Haemophilus_parahaemolyticus Actinobacillus TB       76.0
##  4 Fusobacterium_periodonticum  Fusobacterium  Control  66.0
##  5 Haemophilus_parahaemolyticus Actinobacillus Control  56.1
##  6 uncultured_Streptococcus     Porphyromonas  Control  51
##  7 Prevotella_histicola         Prevotella     Control  46.4
##  8 Veillonella_tobetsuensis     Veillonella    Control  43.2
##  9 <NA>                         Veillonella    Control  38.0
## 10 Sneathia_sanguinegens        Sneathia       TB       37.7
## # i 668 more rows
```

```
merged_df |> dplyr::filter(pipeline == "MetaScope", Sample_type == "Sputum") |>
  dplyr::group_by(Species, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 596 x 3
## # Groups:   Species [596]
##    Species                     status  mean
##    <chr>                       <chr>  <dbl>
##  1 Streptococcus mitis         TB     791.
##  2 Streptococcus salivarius    TB     620.
##  3 Prevotella melaninogenica   TB     103.
##  4 Streptococcus parasanguinis TB      51.2
##  5 Neisseria mucosa            TB      50
##  6 Veillonella parvula         TB      44
##  7 Neisseria sicca             TB      28.3
##  8 Rothia dentocariosa         TB      25
##  9 Streptococcus sanguinis     TB      24.6
## 10 Streptococcus chosunense    TB      23.3
## # i 586 more rows
```

```
merged_df |> dplyr::filter(pipeline == "DADA2", Sample_type == "Sputum") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 455 x 4
## # Groups:   Species, Genus [455]
##    Species                  Genus         status  mean
##    <chr>                    <chr>         <chr>  <dbl>
##  1 Gemella haemolysans      Gemella       TB     240.
##  2 <NA>                     Gemella       TB      68.2
##  3 Moraxella catarrhalis    Moraxella     TB      55
##  4 Streptococcus salivarius Streptococcus TB      33.3
##  5 Rothia mucilaginosa      Rothia        TB      30.6
##  6 Streptococcus mitis      Streptococcus TB      29.6
##  7 Mycobacterium canettii   Mycobacterium TB      27.3
##  8 Neisseria meningitidis   Neisseria     TB      26.7
##  9 Neisseria sicca          Neisseria     TB      26.2
## 10 <NA>                     Neisseria     TB      20.7
## # i 445 more rows
```

```
merged_df |> dplyr::filter(pipeline == "QIIME2", Sample_type == "Sputum") |>
  dplyr::group_by(Species, Genus, status) |>
  dplyr::summarise(mean = mean(Abundance)) |>
  dplyr::arrange(desc(mean))
```

```
## `summarise()` has grouped output by 'Species', 'Genus'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 339 x 4
## # Groups:   Species, Genus [339]
##    Species                   Genus         status  mean
##    <chr>                     <chr>         <chr>  <dbl>
##  1 uncultured_bacterium      Gemella       TB     121.
##  2 Moraxella_catarrhalis     Moraxella     TB      69.7
##  3 Rothia_mucilaginosa       Rothia        TB      52.9
##  4 <NA>                      Rothia        TB      23.5
##  5 Prevotella_melaninogenica Prevotella    TB      23.3
##  6 Leptotrichia_wadei        Leptotrichia  TB      23.3
##  7 Streptococcus_cristatus   Streptococcus TB      22.9
##  8 Schaalia_odontolytica     Actinomyces   TB      22.9
##  9 <NA>                      Mycobacterium TB      21.4
## 10 uncultured_organism       Leptotrichia  TB      20.1
## # i 329 more rows
```

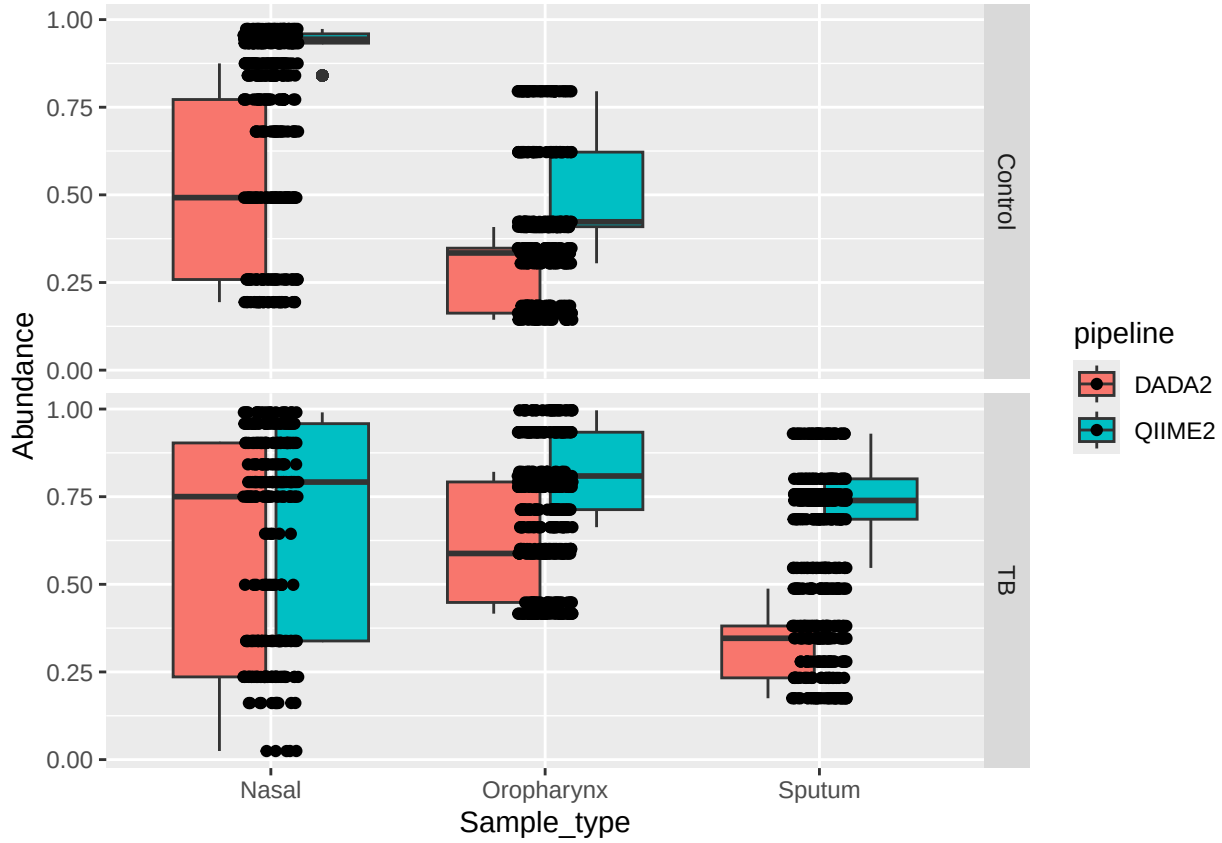
Plotting Unknown Species

```
merged_relab_df |>
  dplyr::filter(is.na(Species) | grepl("uncultured", Species)) |>
  dplyr::filter(Abundance > 0) |>
  dplyr::group_by(Sample, pipeline) |>
  dplyr::summarise(Abundance = sum(Abundance),
                   Sample_type = Sample_type,
```

```
                  status = status) |>
  dplyr::ungroup() |>
  ggplot(aes(x = Sample_type, y = Abundance, fill = pipeline)) +
  geom_boxplot() +
  geom_jitter(width = 0.1) +
  facet_grid(vars(status))
```

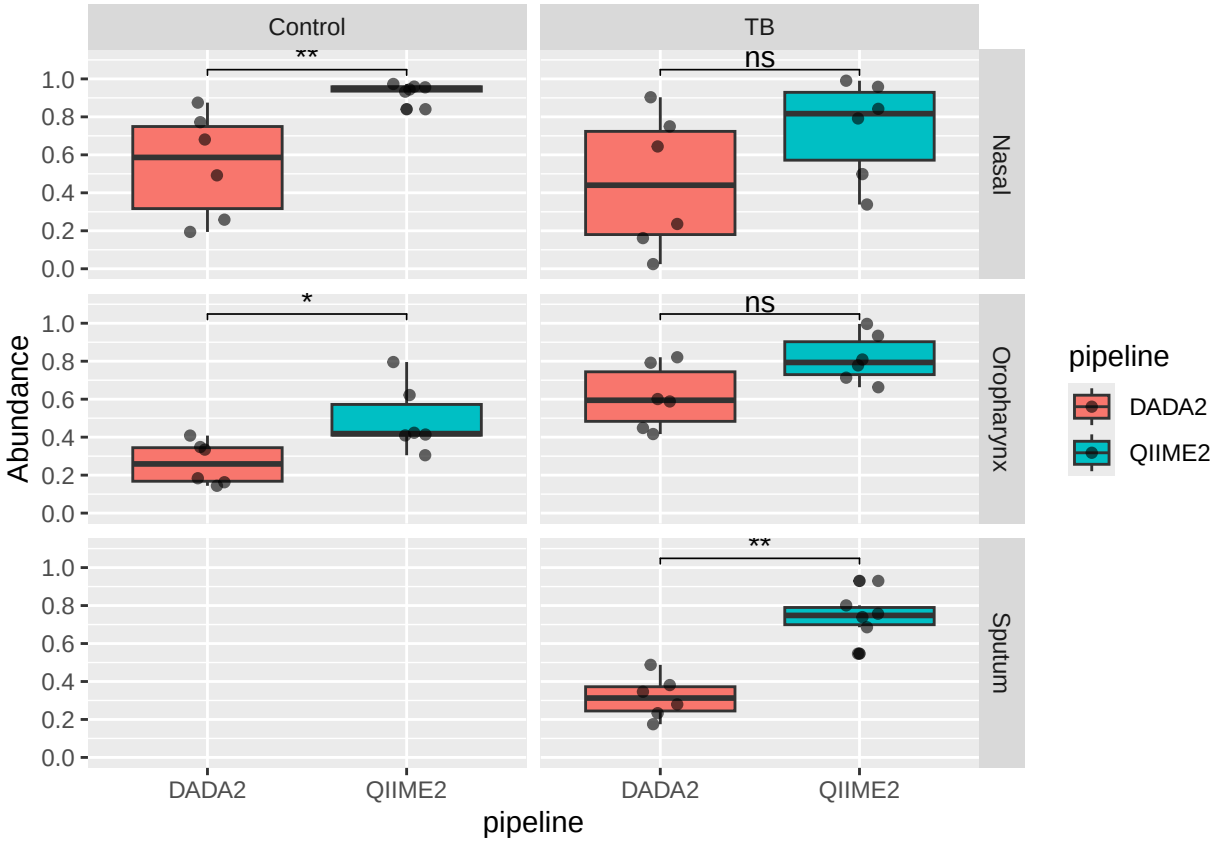
```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'Sample', 'pipeline'. You can override
## using the `.groups` argument.
```




```
merged_relab_df |>
  dplyr::filter(is.na(Species) | grepl("uncultured", Species)) |>
  dplyr::filter(Abundance > 0, pipeline %in% c("DADA2", "QIIME2")) |>
  dplyr::group_by(Sample, pipeline, Sample_type, status) |>
  dplyr::summarise(Abundance = sum(Abundance), .groups = "drop") |>
```

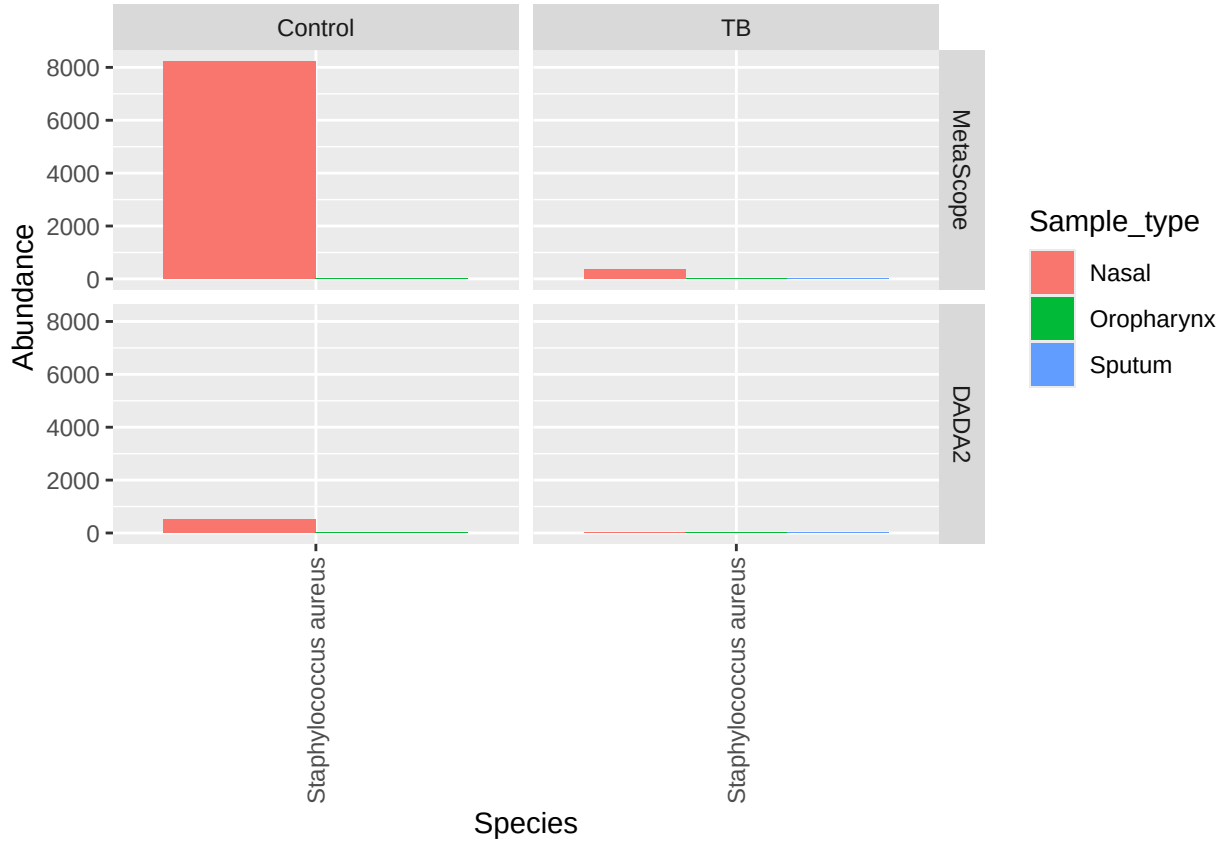
```
  ggplot(aes(x = pipeline, y = Abundance, fill = pipeline)) +
  geom_boxplot(position = position_dodge(width = 0.75)) +
  geom_jitter(
    color = "black",
    position = position_jitterdodge(jitter.width = 0.2, dodge.width = 0.75),
    alpha = 0.6, size = 1.5
  ) +
  scale_y_continuous(breaks = seq(0, 1, 0.2), limits = c(0, 1.1)) +
  stat_compare_means(
    method = "wilcox.test",
    label = "p.signif",
    comparisons = list(c("DADA2", "QIIME2")),
    position = position_dodge(width = 0.75),
    label.y= 1,
  ) +
  facet_grid(rows = vars(Sample_type), cols = vars(status))
```



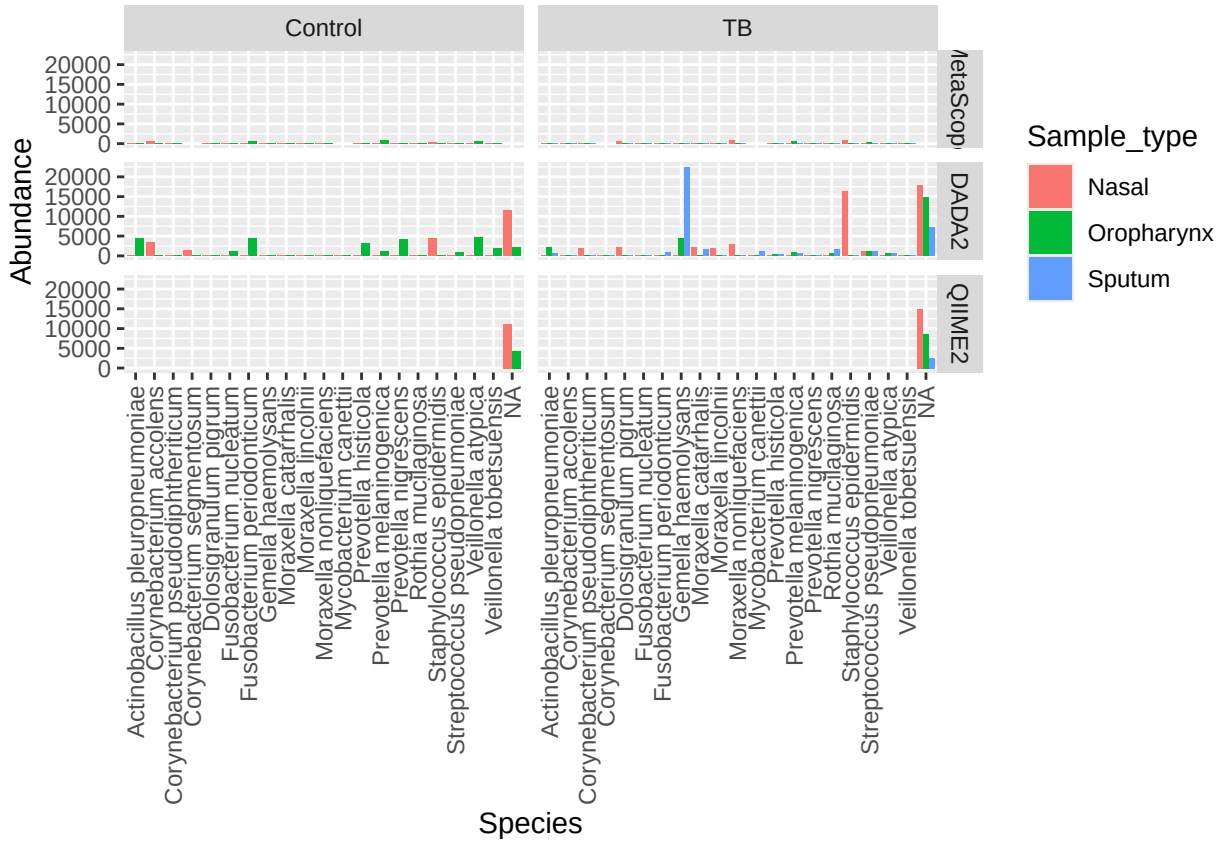
Filtered abundance barplots

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

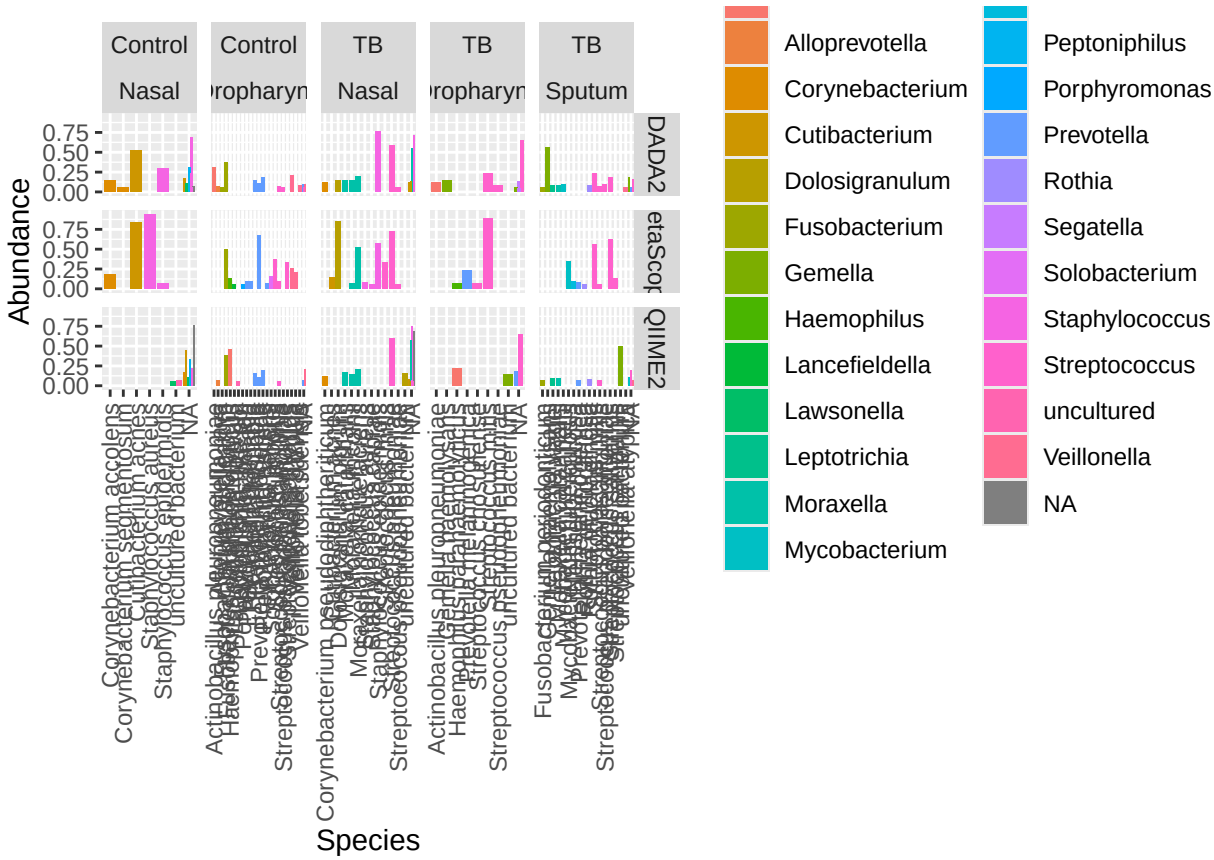
```
high_abund_ms <- merged_df |>
  dplyr::filter(Abundance > 1000) |>
  dplyr::filter(pipeline == "MetaScope")

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
```

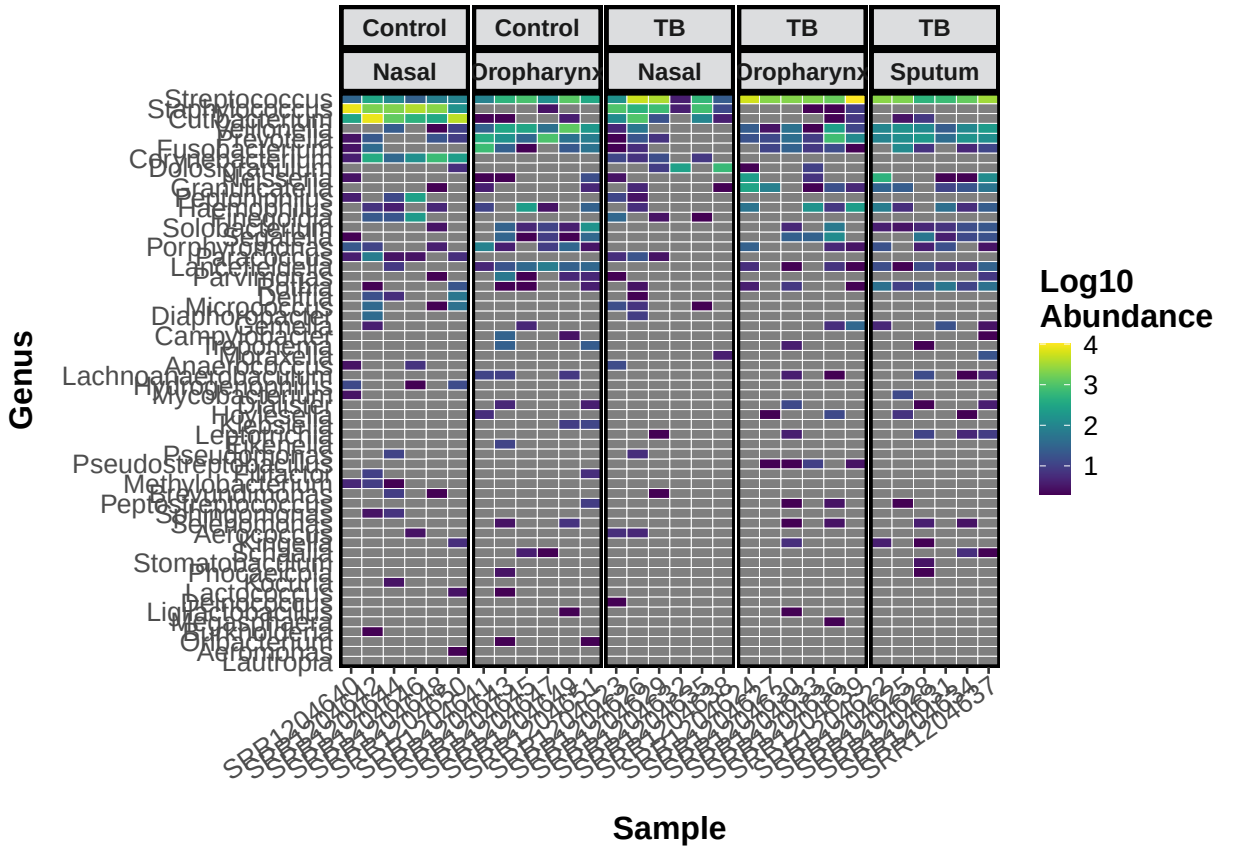



```
high_abund_species <- merged_relab_df |>
  dplyr::filter(Abundance > 0.05) |>
  ggplot(aes(fill=Genus, y=Abundance, x=Species)) +
  geom_bar(position="dodge", stat="identity") +
  facet_grid(rows=vars(pipeline), cols=vars(status,Sample_type),
             scales = "free_x") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

high_abund_species
```

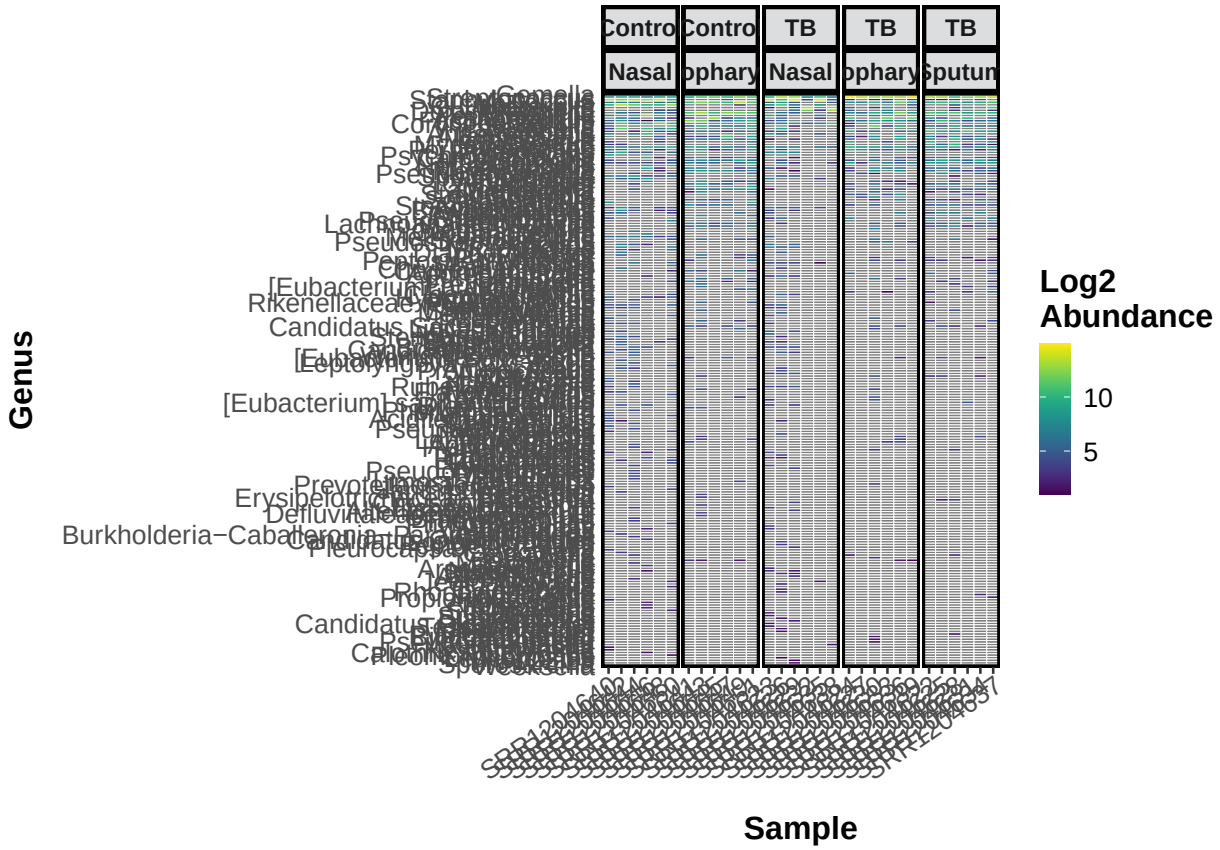


Heatmaps

```
ps_ms_filt <- taxa_filter(ps_ms, frequency = 0.1)

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

```
phyloseq_obj <- ps_ms
treatment = c("status", "Sample_type")
subset = NULL
index = "shannon"

alpha_diversity_table <- function(
  phyloseq_obj,
  treatment = NULL,
  subset = NULL,
  index = "shannon",
  pipeline = NULL)
  {
  phyloseq_obj <-
    taxa_filter(phyloseq_obj, treatment = treatment, subset = subset)
  treatment_name <- paste(treatment, collapse = "_")
  alpha <- data.table::data.table(as(phyloseq_obj@otu_table, "matrix"))
  alpha <- alpha[, lapply(.SD, function(sample) sample / sum(sample))]
  if (index == "shannon") {
    alpha <- -alpha * log(alpha)
  } else {
    alpha <- alpha * alpha
  }
```

```
  alpha <- alpha[, lapply(.SD, sum, na.rm = TRUE)]
  if (index == "simpson") {
    alpha <- 1 - alpha
  } else if (index == "invsimpson") {
    alpha <- 1 / alpha
  }
  graph_data <- data.table::data.table(
    Sample = sample_names(phyloseq_obj),
    Alpha = unlist(alpha)
  )
  graph_data <- merge(graph_data,
    data.table::as.data.table(as(phyloseq_obj@sam_data, "data.frame"),
    keep.rownames = "Sample"), by = "Sample")
  graph_data$pipeline <- pipeline
  return(graph_data)
}

alpha_div_df <- purrr::map2_dfr(
  .x = list(ps_ms, ps_dada2, ps_qiime2),
  .y = c("MetaScope", "DADA2", "QIIME2"),
  .f = ~ alpha_diversity_table(
    phyloseq_obj = .x,
    treatment = c("status", "Sample_type"),
    subset = NULL,
    index = "shannon",
    pipeline = .y))
```

```
clean_richness <- function(ps_obj, pipeline) {
  richness_df <- estimate_richness(ps_obj) |>
    merge(sample_data(ps_obj), by = 0) |>
    mutate(pipeline = pipeline) |>
    pivot_longer(cols = c(Observed, Shannon, Simpson, InvSimpson), names_to = "metric", values_to = "va
    filter(Sample_type != "Sputum")
  return(richness_df)
}
merged_richness <- purrr::map2_dfr(list(ps_ms, ps_dada2, ps_qiime2), list("MetaScope", "DADA2", "QIIME2
```

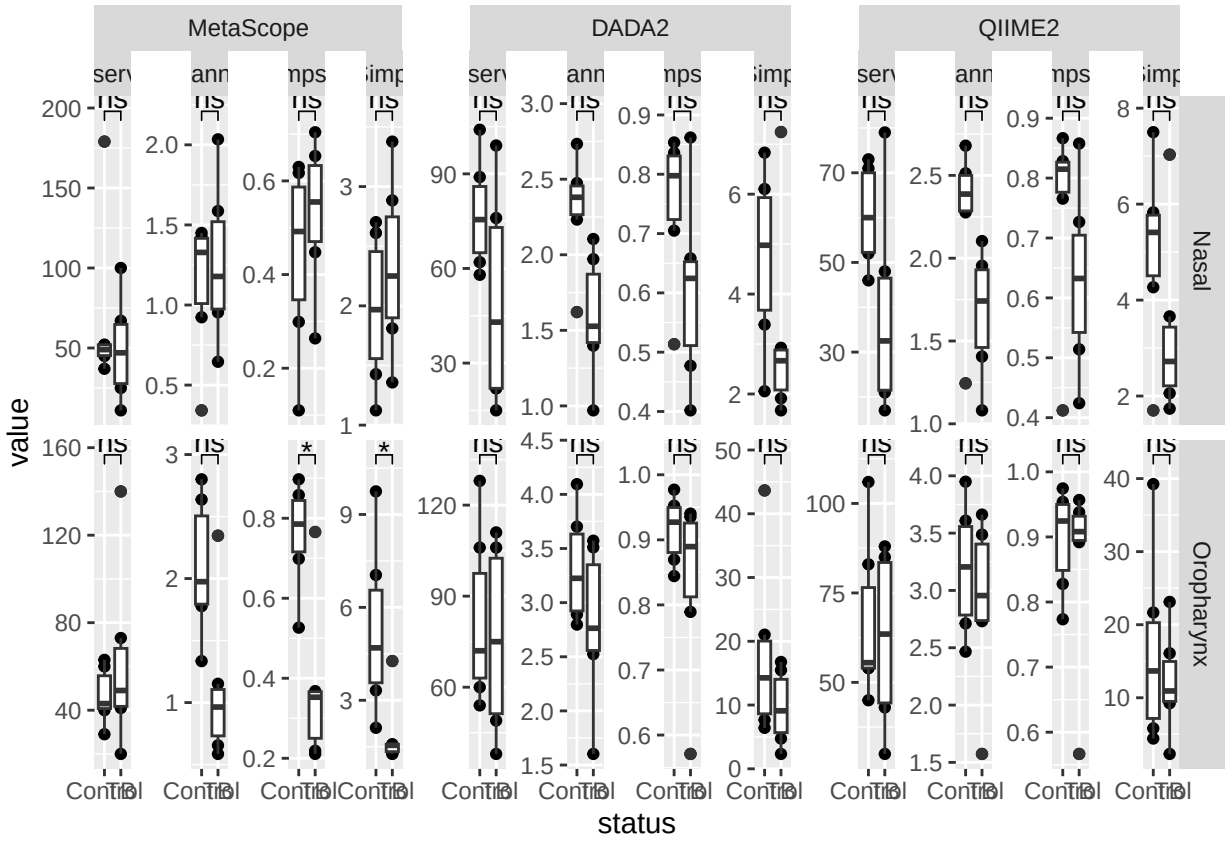
```
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## 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.
```

```
clean_wilcox <- function(ps_obj, pipeline) {
  res <- clean_richness(ps_obj, pipeline) |>
```

```
    group_by(Sample_type, pipeline, metric) |>
    wilcox_test(value ~ status) |>
    add_y_position(fun = "max", step.increase = 0.12,
                   data = group_by(clean_richness(ps_obj, pipeline), Sample_type, pipeline, metric),
                   scales = "free_y") |>
    ungroup() |>
    adjust_pvalue(method = "fdr") |>
    mutate(label = ifelse((p.adj < 0.05), "*", "ns"))
  return(res)
}

merged_wilcox_test <- purrr::map2_dfr(list(ps_ms, ps_dada2, ps_qiime2), list("MetaScope", "DADA2", "QIII
```

```
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## 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.
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## 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.
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## 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.
## Warning in estimate_richness(ps_obj): The data you have provided does not have
## 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.
```

```
merged_richness$metric <- factor(merged_richness$metric, levels = c("Observed", "Shannon", "Simpson", "I
merged_richness$pipeline <- factor(merged_richness$pipeline, levels = c("MetaScope", "DADA2", "QIIME2")
merged_wilcox_test$metric <- factor(merged_wilcox_test$metric, levels = c("Observed", "Shannon", "Simps
merged_wilcox_test$pipeline <- factor(merged_wilcox_test$pipeline, levels = c("MetaScope", "DADA2", "QII
ggplot(merged_richness, aes(x = status, y = value)) +
  geom_point() +
  geom_boxplot() +
  ggh4x::facet_nested(Sample_type ~ pipeline + metric, scales = "free_y", independent = "y") +
  stat_pvalue_manual(mutate(merged_wilcox_test, y.position = y.position * 1.05),
                     label = "label", y.position = "y.position")
```




```
  #scale_y_continuous(expand = expansion(mult = c(0, 0.07)))
```

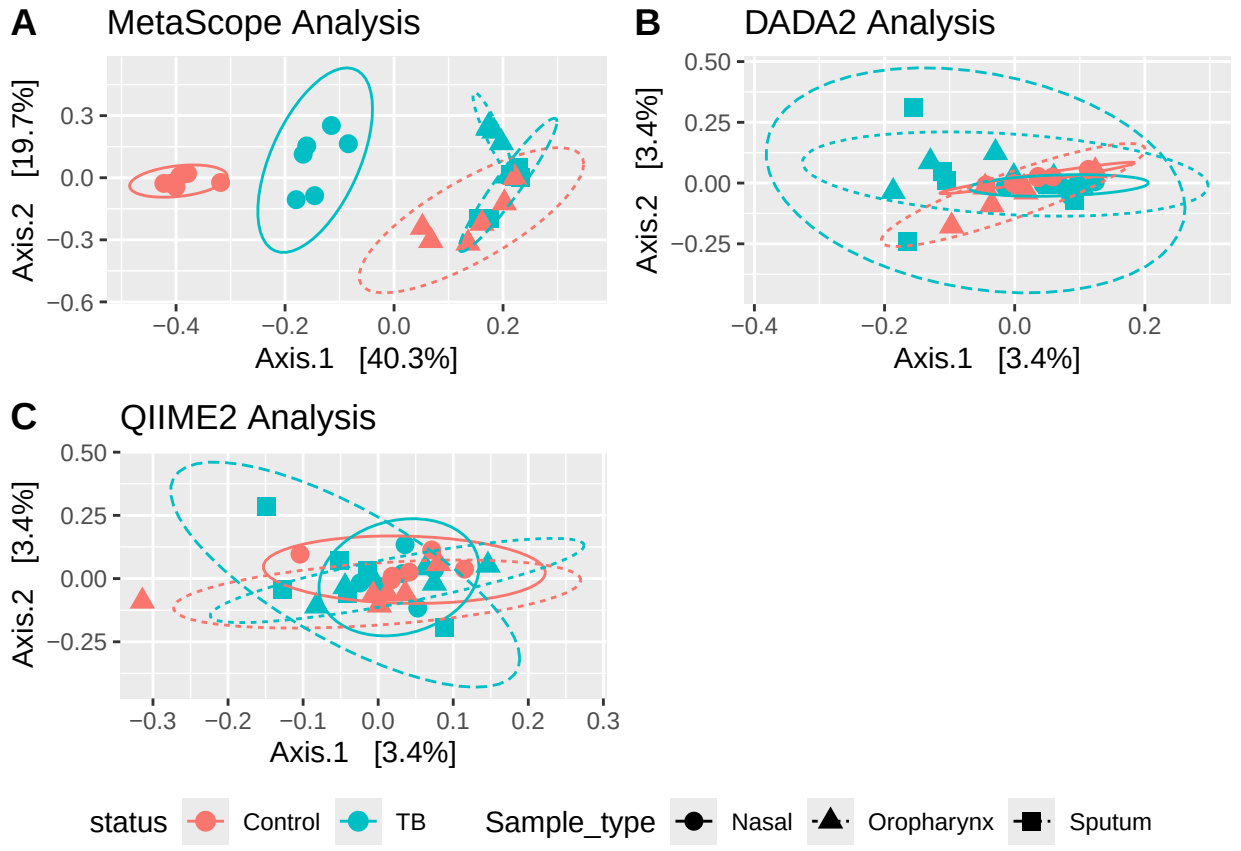
PCOA Plots

```
ps_ms.ord <- ordinate(ps_ms, "PCoA", "jsd")
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", "jsd")
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))

ps_qiime2.ord <- ordinate(ps_qiime2, "PCoA", "jsd")
p3_3 <- plot_ordination(ps_qiime2, ps_qiime2.ord, type="samples", color="status", shape="Sample_type",
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))
```

We use the JSD distance metric because it handles zeros better than bray-curtis distance

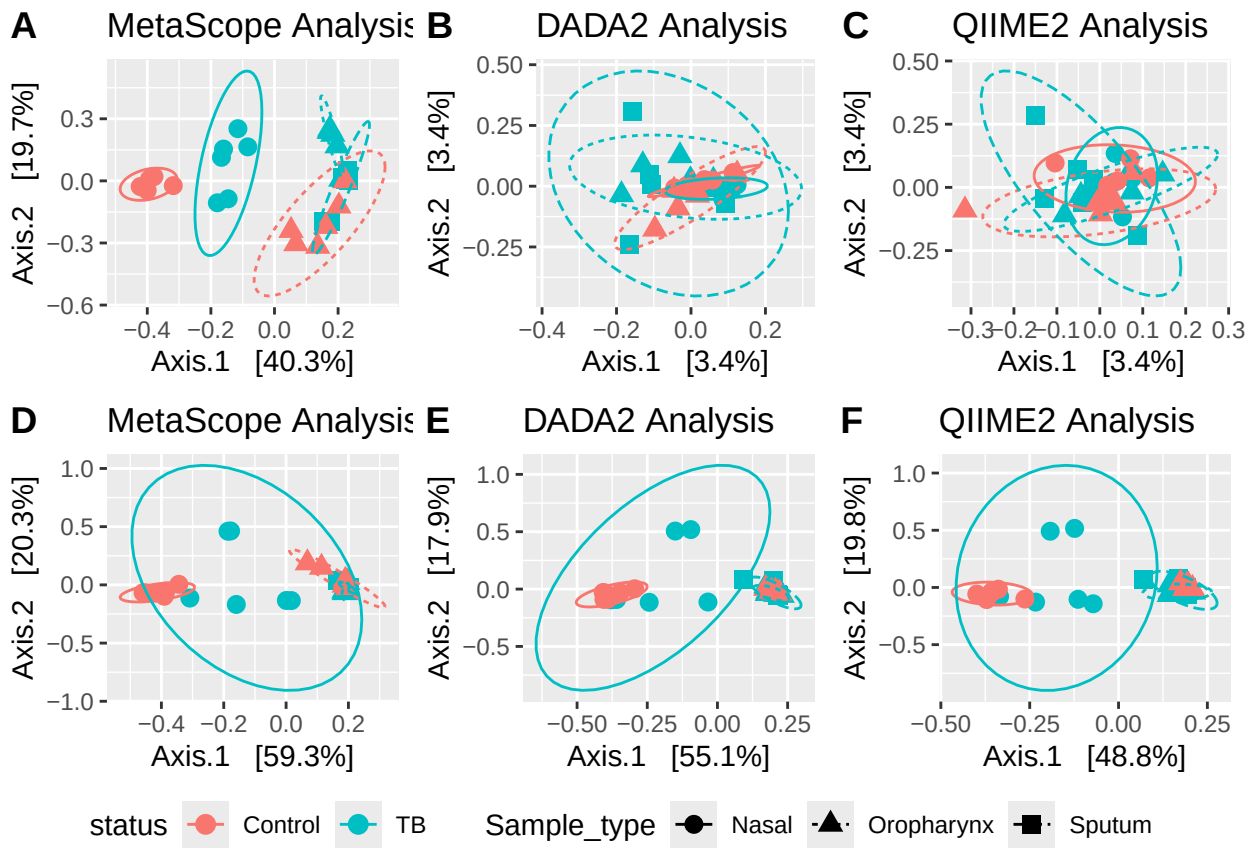
```
ps_ms_genus <- tax_glom(ps_ms, taxrank="Genus")
ps_ms_genus.ord <- ordinate(ps_ms_genus, "PCoA", "jsd")
```

```
p3_4 <- plot_ordination(ps_ms_genus, ps_ms_genus.ord, type="samples", color="status", shape="Sample_type
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))

ps_dada2_genus <- tax_glom(ps_dada2, taxrank="Genus")
ps_dada2_genus.ord <- ordinate(ps_dada2_genus, "PCoA", "jsd")
p3_5 <- plot_ordination(ps_dada2_genus, ps_dada2_genus.ord, type="samples", color="status", shape="Sampl
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))

ps_qiime2_genus <- tax_glom(ps_qiime2, taxrank="Genus")
ps_qiime2_genus.ord <- ordinate(ps_qiime2_genus, "PCoA", "jsd")
p3_6 <- plot_ordination(ps_qiime2_genus, ps_qiime2_genus.ord, type="samples", color="status", shape="Sa
  geom_point(size=3) +
  stat_ellipse(aes(linetype=Sample_type))

ggarrange(p3_1, p3_2, p3_3, labels = "AUTO", common.legend = TRUE, legend = "bottom")
```




```
ggarrange(p3_1, p3_2, p3_3, p3_4, p3_5, p3_6, labels = "AUTO", common.legend = TRUE, legend = "bottom")
```

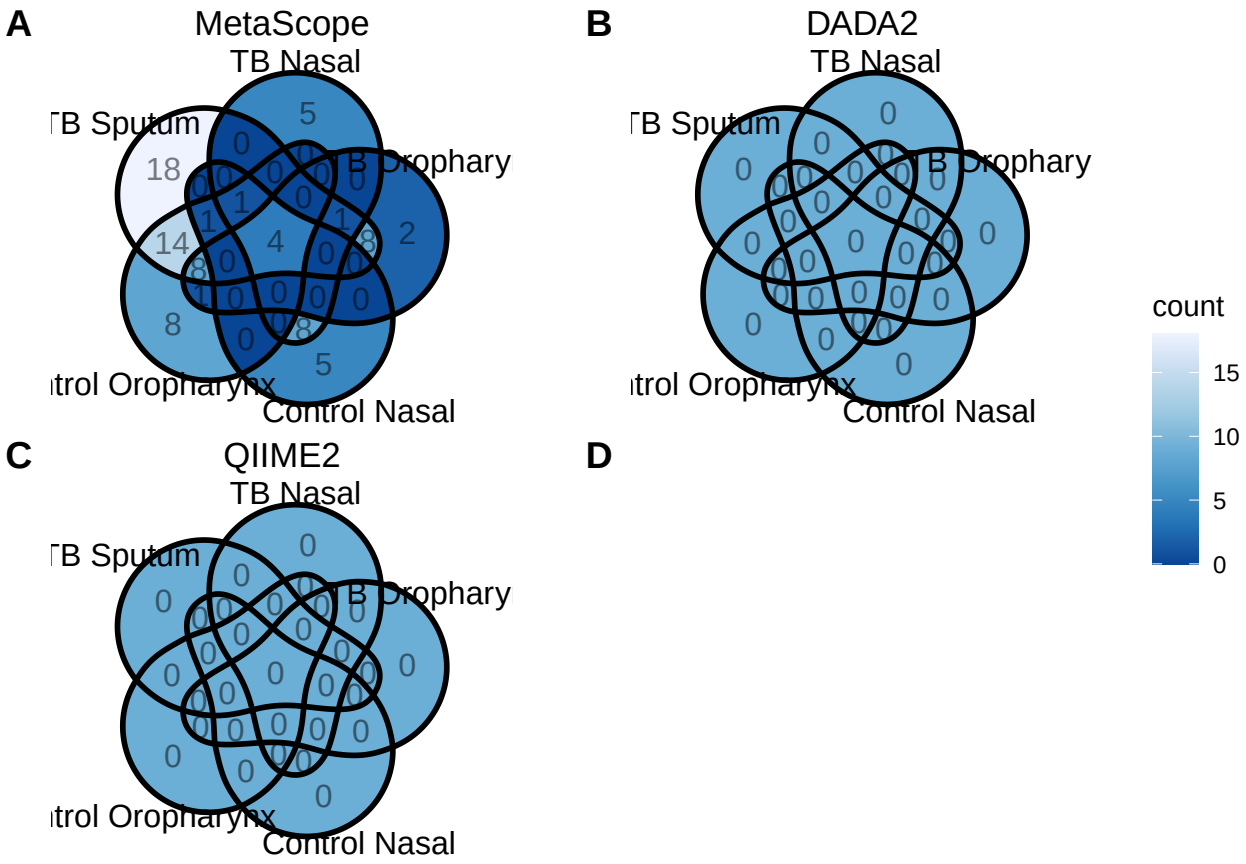
```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```



Core Microbiome Analysis

```
ps_ms_rel <- microbiome::transform(ps_ms, "compositional")
ps_dada2_rel <- microbiome::transform(ps_dada2, "compositional")
ps_qiime2_rel <- microbiome::transform(ps_qiime2, "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") +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
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") +
  theme(plot.title = element_text(hjust = 0.5))

list_core_qiime2<- c() # an empty object to store information
for (i in 1:5){
  ps.sub <- subset_samples(ps_qiime2_rel, status == groups[[i]][1] & Sample_type == groups[[i]][2])
  core_m <- core_members(ps.sub,
                         detection = 0.001,
                         prevalence = 0.2)
  list_core_qiime2[[i]] <- core_m
}
names(list_core_qiime2) <- lapply(groups, function(x) paste(x, collapse = " "))
p4_3 <- ggVennDiagram(list_core_qiime2, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2)) +
  ggtitle("QIIME2") +
  theme(plot.title = element_text(hjust = 0.5))


ggarrange(p4_1, p4_2,p4_3, labels = "AUTO", common.legend = TRUE, legend = "right", nrows = 1)
```




```
ps_ms_rel_genus <- microbiome::transform(tax_glom(ps_ms_rel, taxrank="Genus"), "compositional")
ps_dada2_rel_genus <- microbiome::transform(tax_glom(ps_dada2_rel, taxrank="Genus"), "compositional")
ps_qiime2_rel_genus <- microbiome::transform(tax_glom(ps_qiime2_rel, taxrank="Genus"), "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_genus, 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_4 <- ggVennDiagram(list_core_ms, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2))

list_core_dada <- c() # an empty object to store information
for (i in 1:5){
  ps.sub <- subset_samples(ps_dada2_rel_genus, 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_5 <- ggVennDiagram(list_core_dada, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2))

list_core_qiime2<- c() # an empty object to store information
for (i in 1:5){
  ps.sub <- subset_samples(ps_qiime2_rel_genus, status == groups[[i]][1] & Sample_type == groups[[i]][2]
  core_m <- core_members(ps.sub,
                         detection = 0.001,
                         prevalence = 0.2)
  list_core_qiime2[[i]] <- core_m
}
names(list_core_qiime2) <- lapply(groups, function(x) paste(x, collapse = " "))
p4_6 <- ggVennDiagram(list_core_qiime2, label_geom = "text", label = "count") +
  scale_fill_distiller(palette = "Blues") +
  scale_x_continuous(expand = expansion(mult = .2))


core_microbiome_figure <- ggarrange(p4_1, p4_2, p4_3, p4_4, p4_5, p4_6,
          labels = "AUTO", common.legend = TRUE, legend = "right")
annotate_figure(core_microbiome_figure,
                left = "Genus                                                                    Species
```