

mock_microbiome_analysis

Sean Lu

2025-07-14

Setup

```
library("phyloseq")
library("ggplot2")
library("dplyr")
library("tibble")
library("ggpubr")
library("DESeq2")
library("ggsci")
library("colorspace")
library("flextable")
library("purrr")
library("pracma")
library("tidyverse")
```

Summarize MetaScope Results

```
ground_truth_taxonomy <- read.csv("data_processed/kozich_2013/ground_truth_taxonomy.csv",
                                   head = TRUE)
species_ground_truth <- ground_truth_taxonomy$species
accessions_path <- "reflib/accessionTaxa.sql"

## Code used to generate ground truth taxids
#ground_truth_taxids <- ground_truth_taxonomy |>
# tidyr::pivot_longer(everything(), names_to = "Taxonomy Level", values_to = "taxon") |>
# dplyr::distinct(taxon, .keep_all = TRUE) |>
# mutate(taxid = taxonomizr::getId(taxon, sqlFile = accessions_path)) |>
# rowwise() |>
# mutate(taxid = as.numeric(strsplit(taxid, split = ",")[[1]][1]))
#write.csv(ground_truth_taxids,
#          file = "data_processed/kozich_2013/ground_truth_taxids.csv",
#          row.names = FALSE)
ground_truth_taxids <- read.csv("data_processed/kozich_2013/ground_truth_taxids.csv",)

# Function used to read MetaScope outputs and generate cleaned accuracy metrics
create_summary_df <- function(csv_paths, pipeline_name) {
  summary_df <- data.frame(
    Species = c("No Call", "Incorrect Call", species_ground_truth))
```

```

n = 1
for (i in csv_paths) {
  # Get Sample Names
  sample_name <- sub("\\.*$", "", basename(i))
  # Clean MetaBlast Table
  metascope_df <- read.csv(i, head = TRUE)
  head(metascope_df)
  metascope_df <- metascope_df |>
    dplyr::select(Genome, read_count) |>
    dplyr::mutate_if(is.numeric, ~ . / sum(.))
  colnames(metascope_df) <- c("Species", sample_name)
  summary_df <- dplyr::left_join(summary_df, metascope_df, by = "Species")
  summary_df[is.na(summary_df)] <- 0

  no_call.metascope <- metascope_df |>
    dplyr::filter(is.na(Species)) |>
    dplyr::select(sample_name) |>
    sum()
  correct_call.metascope <- metascope_df |>
    dplyr::filter(Species %in% species_ground_truth) |>
    dplyr::select(sample_name) |>
    sum()
  incorrect_call.metascope <- 1 - correct_call.metascope - no_call.metascope

  summary_df[1,n+1] <- no_call.metascope
  summary_df[2,n+1] <- incorrect_call.metascope
  n = n + 1
}

summary_df_long <- tidyr::pivot_longer(
  summary_df,
  cols = c(2:ncol(summary_df)),
  values_to = "prop"
)
summary_df_long$Species <- factor(summary_df_long$Species,
  levels = c("No Call", "Incorrect Call", species_ground_truth))
summary_df_long$name <- factor(summary_df_long$name)
summary_df_long$pipeline <- factor(rep(pipeline_name, nrow(summary_df_long)))
return(summary_df_long)
}

## Generate Summary dataframes for all metascope outputs
ms_df <- create_summary_df(list.files(path = "data_processed/kozich_2013/results",
  pattern = ".metascope_id.csv",
  full.names = TRUE,
  recursive = TRUE),
  pipeline = "MetaScope")

```

```

## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
## # Was:
## data %>% select(sample_name)

```

```
##
## # Now:
## data %>% select(all_of(sample_name))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

ms_p_df <- create_summary_df(list.files(path = "data_processed/kozich_2013/results_priors",
                                     pattern = ".metascope_id.csv",
                                     full.names = TRUE,
                                     recursive = TRUE),
                           pipeline = "MetaScope Priors")
ms_p_b_df <- create_summary_df(list.files(path = "data_processed/kozich_2013/results_metablast_priors_1.",
                                     pattern = ".metascope_id.csv",
                                     full.names = TRUE,
                                     recursive = TRUE),
                           pipeline = "MetaBlast")
summary_df <- rbind(ms_df, ms_p_df, ms_p_b_df)

## Add Ground Truth to summary dataframe
ground_truth_df <- data.frame(
  Species = c("No Call", "Incorrect Call", species_ground_truth))
ground_truth_df <- cbind(ground_truth_df,
                        as.data.frame(do.call(cbind, replicate(33, c(0,0,rep(1/21, 21))), simplify = FALSE)),
                        colnames(ground_truth_df) <- c("Species", as.character(unique(summary_df$name)))
ground_truth_df <- tidyr::pivot_longer(
  ground_truth_df,
  cols = c(2:34),
  values_to = "prop"
)
ground_truth_df$pipeline <- "Ground Truth"
summary_df <- rbind(summary_df, ground_truth_df)
```

Summarize DADA2 Results

```
##Left joining with aggregate because duplicate species names
dada2_files <- list.files(path = "data_processed/kozich_2013/dada2_results",
                        full.names = TRUE)
taxonomy_cols <- c("Species", "Genus", "Family", "Order", "Class", "Phylum", "Kingdom")

## Join DADA2 Files together
dada2_df <- purrr::map_dfr(dada2_files, function(x){
  sample_name <- sub("\\\\.*$", "", basename(x)) |> strsplit(split = "dada2_")
  sample_name <- sample_name[[1]][2]
  res <- read.csv(x) |>
    dplyr::mutate(Species = ifelse(is.na(Species), NA, paste0(Genus, " ", Species))) |>
    dplyr::mutate_if(is.numeric, ~ . / sum(.)) |>
    dplyr::mutate(pipeline = "DADA2-NB", name = sample_name) |>
    dplyr::rowwise() |>
```

```

dplyr::mutate(
  taxon = {
    tax_vals <- c_across(all_of(taxonomy_cols))
    non_na_vals <- tax_vals[!is.na(tax_vals)]
    if (length(non_na_vals) > 0) non_na_vals[1] else NA_character_
  }
) |>
dplyr::ungroup()
})

## Clean up old taxonomy names
dada2_df$taxon[dada2_df$taxon == "Bacteroides vulgatus"] <- 'Phocaeicola vulgatus'
dada2_df$taxon[dada2_df$taxon == 'Actinomyces odontolyticus'] <- 'Schaalia odontolytica'
dada2_df$taxon[dada2_df$taxon == "Propionibacterium acnes"] <- 'Cutibacterium acnes'
dada2_df$taxon[dada2_df$taxon == "Rhodobacter sphaeroides"] <- 'Cereibacter sphaeroides'
dada2_df$taxon[dada2_df$taxon == "Clostridium sensu stricto 1 beijeinckii"] <- 'Clostridium beijeinckii'
dada2_df$taxon[dada2_df$taxon == "Escherichia-Shigella"] <- 'Escherichia' # Assuming DADA2 meant Escher
dada2_df$taxon[dada2_df$taxon == "Prevotella_9"] <- 'Segatella'
dada2_df$taxon[dada2_df$taxon == "Prevotella_9 copri"] <- 'Segatella corpi'
dada2_df$taxon[dada2_df$taxon == "Rhodobacteraceae"] <- 'Paracoccaceae'
dada2_df$taxon[dada2_df$taxon == "Bacteroides massiliensis"] <- 'Phocaeicola massiliensis'
dada2_df$taxon[dada2_df$taxon == "Ruminococcaceae"] <- 'Oscillospiraceae'

## Add NCBI taxonomy ids
dada2_df <- dada2_df |>
  mutate(taxid_raw = taxonomizr::getId(taxon, sqlFile = accessions_path)) |>
  rowwise() |>
  mutate(taxid = as.numeric(strsplit(taxid_raw, split = ",")[[1]][1])) |>
  ungroup() |>
  mutate(taxid = if_else(is.na(taxid), 9999999, taxid)) |> # REPLACING ALL UNKNOWN TAXONS WITH TAXID 99
dplyr::select(!taxid_raw)

```

```

## Warning: There was 1 warning in 'mutate()'.
## i In argument: 'taxid_raw = taxonomizr::getId(taxon, sqlFile =
##   accessions_path)'.
## Caused by warning in 'taxonomizr::getId()':
## ! Multiple taxa ids found for Bacillus, Serratia. Collapsing with commas

```

```

## Adding categories
dada2_df_cats <- dada2_df |>
  dplyr::mutate(category = case_when(
    (taxid %in% ground_truth_taxids$taxid) & !is.na(Species) ~ taxon,
    (taxid %in% ground_truth_taxids$taxid) & is.na(Species) ~ "No Call",
    !(taxid %in% ground_truth_taxids$taxid) ~ "Incorrect Call")) |>
  dplyr::group_by(category, name) |>
  dplyr::summarise(prop = sum(reads_count))

```

```

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

```

```

dada2_df_cats$pipeline = "DADA2-NB"
colnames(dada2_df_cats) <- c("Species", "name", "prop", "pipeline")

```

```
#write.csv(dada2_df_cats, file = "data_processed/kozich_2013/dada2_df.csv",
#          row.names = FALSE)

summary_df <- rbind(summary_df, dada2_df_cats)
```

Summarize QIIME2 Results

```
summary_df_qiime2 <- data.frame(
  Species = c("No Call", "Incorrect Call", species_ground_truth))

qiime2_otu_table <- read.table("data_processed/kozich_2013/qiime2_results/feature_table.tsv",
  header = TRUE, sep = "\t") |>
  dplyr::mutate_if(is.numeric, ~ . / sum(.)) |>
  pivot_longer(cols = !OTU.ID)

taxonomy_cols <- c("Species", "Genus", "Family", "Order", "Class", "Phylum", "Superkingdom")
qiime2_tax_table <- read.table("data_processed/kozich_2013/qiime2_results/taxonomy.tsv",
  header = TRUE, sep = "\t") |>
  separate(Taxon, into = c("Superkingdom", "Phylum", "Class",
    "Order", "Family", "Genus", "Species"),
    sep = ";", fill = "right", extra = "drop") |>
  mutate(across(Superkingdom:Species, ~ str_remove(., "^\\s*[a-z]__"))) |>
  mutate(Species = sub("_", " ", Species)) |>
  rowwise() |>
  mutate(
    taxon = {
      tax_vals <- c_across(all_of(taxonomy_cols))
      non_na_vals <- tax_vals[!is.na(tax_vals)]
      if (length(non_na_vals) > 0) non_na_vals[1] else NA_character_0}) |>
  ungroup()

# Manually Address NAs
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Clostridium_sensu_stricto_1"] <- "Clostridium"
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Escherichia-Shigella"] <- "Escherichia" # Escherichia
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Bacteroides vulgatus"] <- "Phocaeicola vulgatus"
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Bacteroides massiliensis"] <- "Phocaeicola massiliensis"
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Alistipes obesi"] <- "Alistipes communis"
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Candidatus_Udaeobacter"] <- "Candidatus Udaeobacter"
qiime2_tax_table$taxon[qiime2_tax_table$taxon == "Ruminococcaceae"] <- "Oscillospiraceae"

# Add taxids to QIIME2 taxonomy table
qiime2_tax_table <- qiime2_tax_table |>
  mutate(taxid_raw = taxonomizr::getId(taxon, sqlFile = accessions_path)) |>
  rowwise() |>
  mutate(taxid = as.numeric(strsplit(taxid_raw, split = ",")[[1]][1])) |>
  ungroup() |>
  mutate(taxid = if_else(is.na(taxid), 9999999, taxid)) |> # REPLACING ALL UNKNOWN TAXONS WITH TAXID 9999999
  dplyr::select(!taxid_raw)
```

```
## Warning: There was 1 warning in 'mutate()'.
```

```
## i In argument: 'taxid_raw = taxonomizr::getId(taxon, sqlFile =
##   accessions_path)'.
## Caused by warning in 'taxonomizr::getId()':
## ! Multiple taxa ids found for Bacillus. Collapsing with commas
```

```
qiime2_df <- full_join(qiime2_otu_table, qiime2_tax_table, by=join_by(OTU.ID == Feature.ID))
```

```
qiime2_df_cats <- qiime2_df |>
  dplyr::mutate(category = case_when(
    (taxid %in% ground_truth_taxids$taxid) & !is.na(Species) ~ taxon,
    (taxid %in% ground_truth_taxids$taxid) & is.na(Species) ~ "No Call",
    !(taxid %in% ground_truth_taxids$taxid) ~ "Incorrect Call")) |>
  dplyr::group_by(category, name) |>
  dplyr::summarise(prop = sum(value))
```

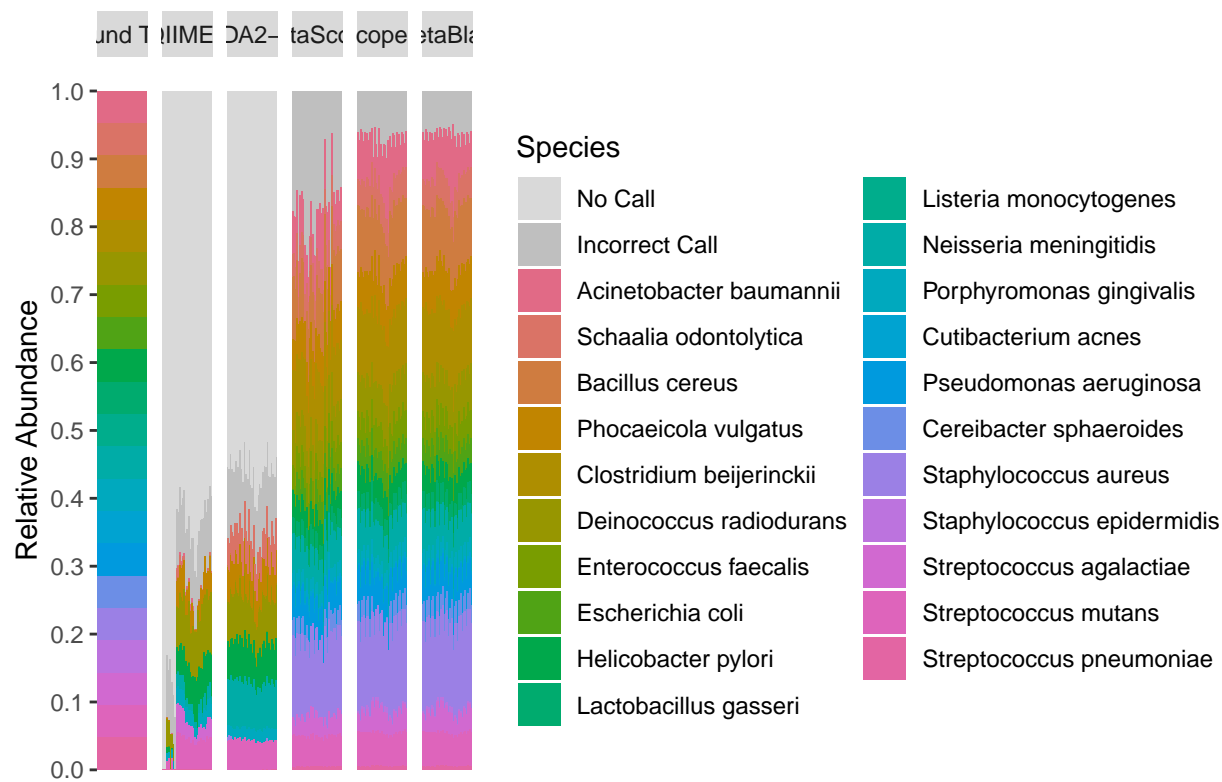
```
## 'summarise()' has grouped output by 'category'. You can override using the
## '.groups' argument.
```

```
qiime2_df_cats$pipeline <- "QIIME2"
colnames(qiime2_df_cats) <- c("Species", "name", "prop", "pipeline")

summary_df_qiime2 <-
  expand_grid(Species = c("No Call", "Incorrect Call", species_ground_truth),
             name = unique(qiime2_df_cats$name),
             stringsAsFactors = TRUE) |>
  merge(qiime2_df_cats, by = c("Species", "name"), all.x = TRUE) |>
  replace_na(list(prop = 0, pipeline = "QIIME2"))

#write.csv(summary_df_qiime2, file = "data_processed/kozich_2013/qiime2_df.csv",
#          row.names = FALSE)
summary_df <- rbind(summary_df, summary_df_qiime2)
```

Plotting Relative Abundance of Mock Microbiome



```
## Warning: fonts used in 'flextable' are ignored because the 'pdflatex' engine is
## used and not 'xelatex' or 'lualatex'. You can avoid this warning by using the
## 'set_flextable_defaults(fonts_ignore=TRUE)' command or use a compatible engine
## by defining 'latex_engine: xelatex' in the YAML header of the R Markdown
## document.
```

Profiler	Correct Call	No Call	Incorrect Call
QIIME2	0.211 ± 0.116	0.708 ± 0.135	0.081 ± 0.028
DADA2-NB	0.347 ± 0.023	0.56 ± 0.023	0.093 ± 0.006
MetaScope	0.821 ± 0.047	0 ± 0	0.179 ± 0.047
MetaScope Priors	0.929 ± 0.013	0 ± 0	0.071 ± 0.013
MetaBlast	0.936 ± 0.009	0 ± 0	0.064 ± 0.009

```
# This metascope is k - 25
p_bar_with_stats<- summary_df |> dplyr::filter(Species %in% c("No Call", "Incorrect Call")) |>
  tidyr::pivot_wider(names_from = Species, values_from = prop, names_repair = "universal") |>
  dplyr::mutate(Correct.Call = 1 - (No.Call + Incorrect.Call)) |>
  dplyr::filter(pipeline != "Ground Truth") |>
  dplyr::group_by(pipeline) |>
```

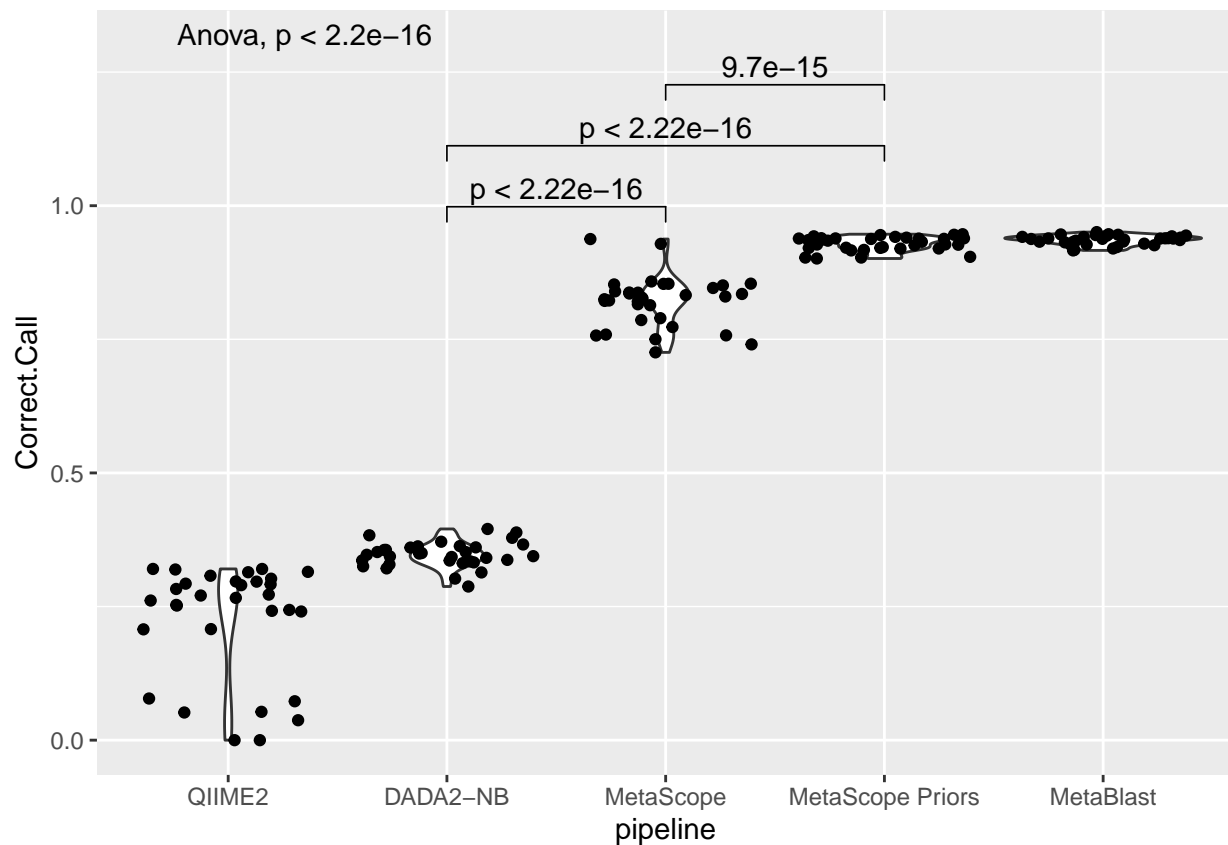
```
dplyr::group_by(name) |>
ggplot(aes(x = pipeline, y = Correct.Call)) +
  geom_violin() +
  geom_jitter() +
  ylim(0,1.3) +
  stat_compare_means(label.y = 1.3, method = "anova", paired = TRUE) +
  stat_compare_means(comparisons = list(c("DADA2-NB", "MetaScope"),
                                         c("DADA2-NB", "MetaScope Priors"),
                                         c("MetaScope", "MetaScope Priors"),
                                         c("MetaScope Priors", "MetaBlast")),
                    method = "t.test",
                    paired = TRUE)
```

```
## New names:
## * 'No Call' -> 'No.Call'
## * 'Incorrect Call' -> 'Incorrect.Call'
```

```
p_bar_with_stats
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## ('geom_signif()').
```



Sensitivity Analysis

```
priors_0<- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "No Priors"
)

priors_0.005 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_0.005",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "0.005"
)

priors_0.01 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_0.01",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "0.01"
)

priors_0.05 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_0.05",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "0.05"
)

priors_0.1 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_0.1",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "0.1"
)

priors_0.5 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_0.5",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
```

```

    pipeline_name = "0.5"
  )

priors_1.0 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_1.0",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "1.0"
)

priors_2.0 <- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_priors_2.0",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "2.0"
)

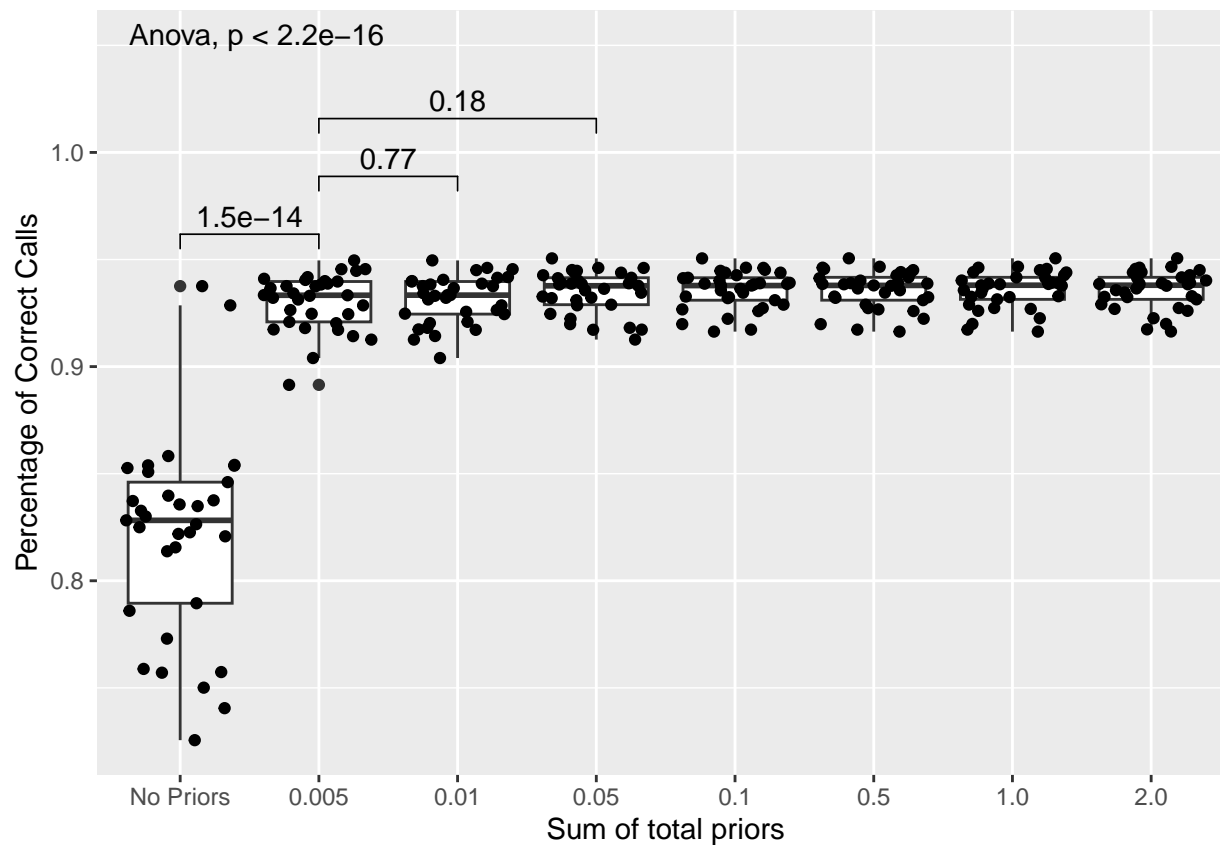
summary_df_sensitivity <- rbind(priors_0, priors_0.005, priors_0.01, priors_0.05, priors_0.1, priors_0.5)

summary_df_sensitivity |> dplyr::filter(Species %in% c("Correct Call", "Incorrect Call")) |>
  tidyr::pivot_wider(names_from = Species, values_from = prop, names_repair = "universal") |>
  dplyr::mutate(Correct.Call = 1 - as.numeric(Incorrect.Call)) |>
  dplyr::filter(pipeline != "Ground Truth") |>
  dplyr::group_by(pipeline) |>
  dplyr::group_by(name) |>
  ggplot(aes(x = pipeline, y = Correct.Call)) +
  geom_boxplot() +
  geom_jitter() +
  #theme(axis.text.x = element_text(angle = 90)) +
  xlab("Sum of total priors") +
  ylab("Percentage of Correct Calls") +
  stat_compare_means(label.y = 1.05, method = "anova", paired = TRUE) +
  stat_compare_means(comparisons = list(c("No Priors", "0.005"),
                                         c("0.005", "0.01"), c("0.005", "0.05")),
                    method = "wilcox.test",
                    paired = FALSE)

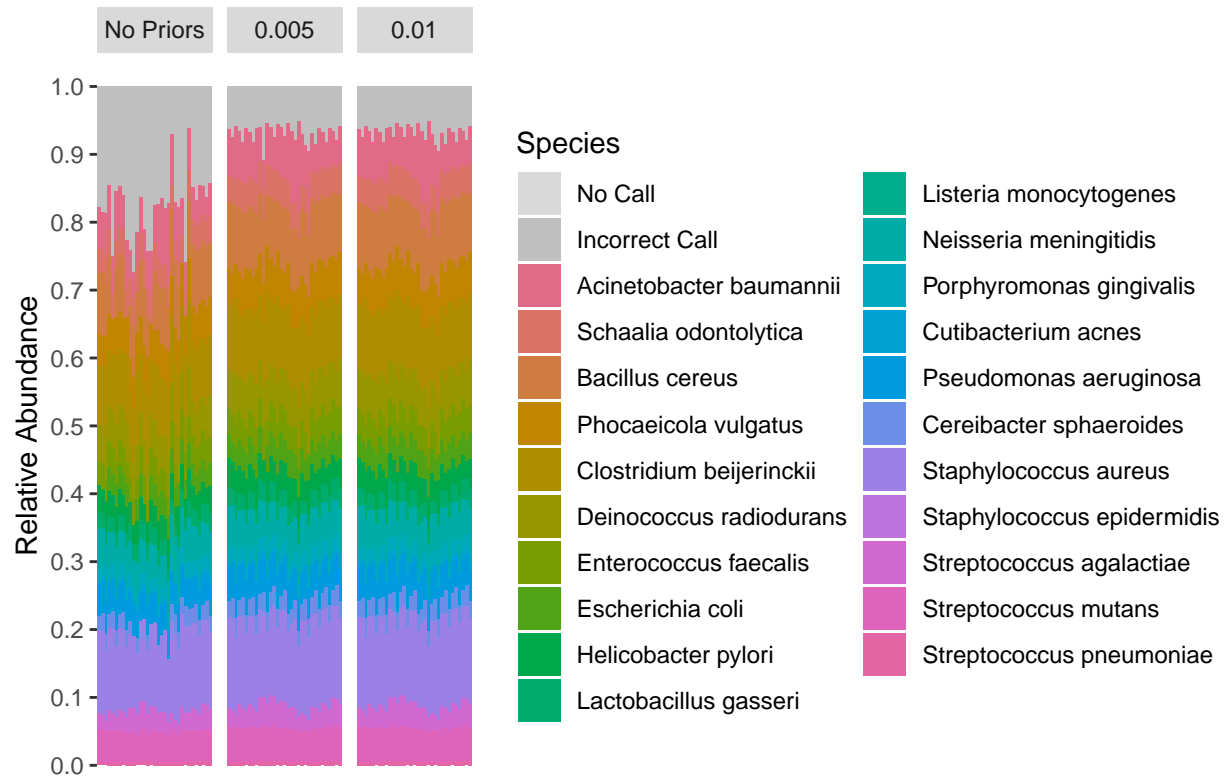
## New names:
## * 'Incorrect Call' -> 'Incorrect.Call'

## Warning in wilcox.test.default(c(0.936613634323669, 0.92448846455617,
## 0.941782795512704, : cannot compute exact p-value with ties

```



```
summary_df_final_3 <- rbind(priors_0, priors_0.005, priors_0.01)
p3 <- ggplot(data = summary_df_final_3, aes(fill = Species, y = prop, x = name)) +
  geom_bar(position = "stack", stat = "identity") +
  scale_fill_manual(values = wheel_colors, name = "Species") +
  ylab("Relative Abundance") +
  xlab("") +
  scale_y_continuous(breaks = seq(0, 1, by = 0.1)) +
  facet_grid(~pipeline) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank())
plot(p3)
```



#HMP priors

```
hmp_priors<- create_summary_df(
  list.files(
    path = "data_processed/kozich_2013/results_hmp_priors",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline_name = "MetaScope HMP Priors"
)

hmp_test <- rbind(priors_0, hmp_priors, priors_1.0) |>
  dplyr::filter(Species %in% c("No Call", "Incorrect Call")) |>
  tidyr::pivot_wider(names_from = Species, values_from = prop, names_repair = "universal") |>
  dplyr::mutate(Correct.Call = 1 - (No.Call + Incorrect.Call))
```

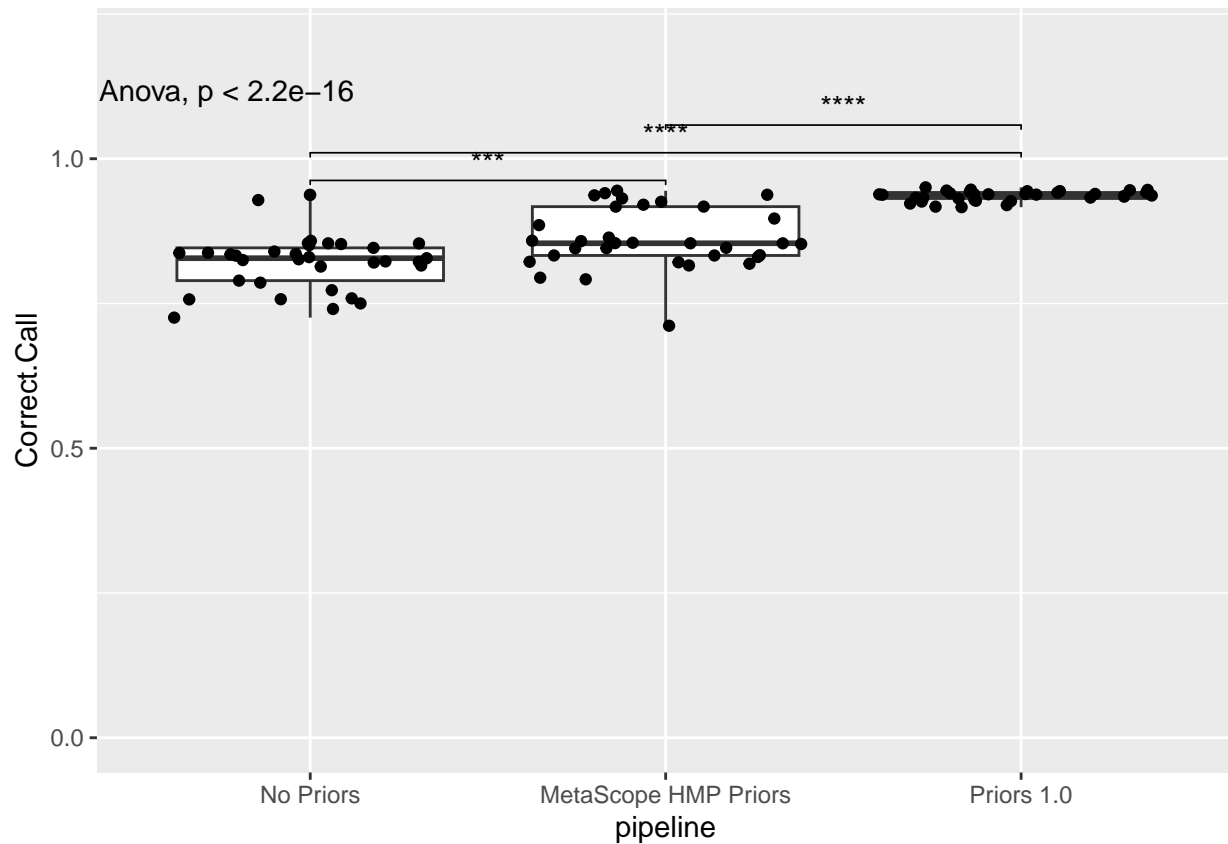
```
## New names:
## * 'No Call' -> 'No.Call'
## * 'Incorrect Call' -> 'Incorrect.Call'
```

```
ggplot(hmp_test, aes(x=pipeline, y=Correct.Call)) +
  geom_boxplot() +
  stat_compare_means(label.y = 1.1, label.x = 0.55, method = "anova", paired = TRUE) +
  stat_compare_means(comparisons = list(c("No Priors", "MetaScope HMP Priors"),
```

```

c("No Priors", "1.0"), c("MetaScope HMP Priors", "1.0")),
method = "wilcox.test",
paired = TRUE,
step.increase = .2,
symnum.args = list(cutpoints = c(0, 0.0001, 0.001, 0.01, 0.05, 1),
symbols = c("****", "****", "***", "*", "ns"))) +
geom_jitter() +
scale_x_discrete(labels=c("1.0" = "Priors 1.0")) +
scale_y_continuous(breaks=c(0,0.5,1),
limits = c(0,1.2))

```



```

accessions_path <- "reflib/accessionTaxa.sql"

metascope_generate_ps_object <- function(file_paths, pipeline) {
  dfs_list <- file_paths |>
  set_names(function(x) sub("\\..*$", "", basename(x))) |>
  map(function(x) {
    df <- read.csv(x)
    sample_name <- sub("\\..*$", "", basename(x))
    df <- df |>
    select(TaxonomyID, Genome, readsEM) |>
    dplyr::rename(!paste0(sample_name, ".", pipeline) := readsEM)
    return(df)
  })
  dfs_merged <- reduce(dfs_list, full_join, by = c("TaxonomyID", "Genome")) |>

```

```

    mutate(across(where(is.numeric), ~ replace_na(.x, 0))) |>
    dplyr::select(-Genome)
otu_mat <- dfs_merged |>
  tibble::column_to_rownames("TaxonomyID") |>
  as.matrix()
tax_mat <- dfs_merged$TaxonomyID |> taxonomizr::getTaxonomy(sqlFile = accessions_path) |>
  as.matrix()
return(list(otu_mat, tax_mat))
}
ms_mats <- metaspope_generate_ps_object(
  list.files(path = "data_processed/kozich_2013/results",
    pattern = ".metaspope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
  pipeline = "MetaScope")

ground_truth_tax_mat <- ground_truth_taxonomy |>
  mutate(taxid = taxonomizr::getId(species, accessions_path)) |>
  tibble::column_to_rownames("taxid")

ground_truth_otu_mat <- matrix(1/21, nrow = 21, ncol = 33,
  dimnames = list(rownames(ground_truth_tax_mat),
    lapply(unique(ground_truth_df$name), paste0, ".", "ground_truth_"))
  as.data.frame() |>
  tibble::rownames_to_column("taxid")

merged_otu_mat <- ms_mats[[1]] |>
  as.data.frame() |>
  mutate(across(everything(), ~ .x / sum(.x))) |>
  tibble::rownames_to_column("taxid") |>
  full_join(ground_truth_otu_mat, by = "taxid") |>
  tibble::column_to_rownames("taxid") |>
  mutate(across(everything(), ~ replace_na(.x, 0))) |>
  as.matrix()

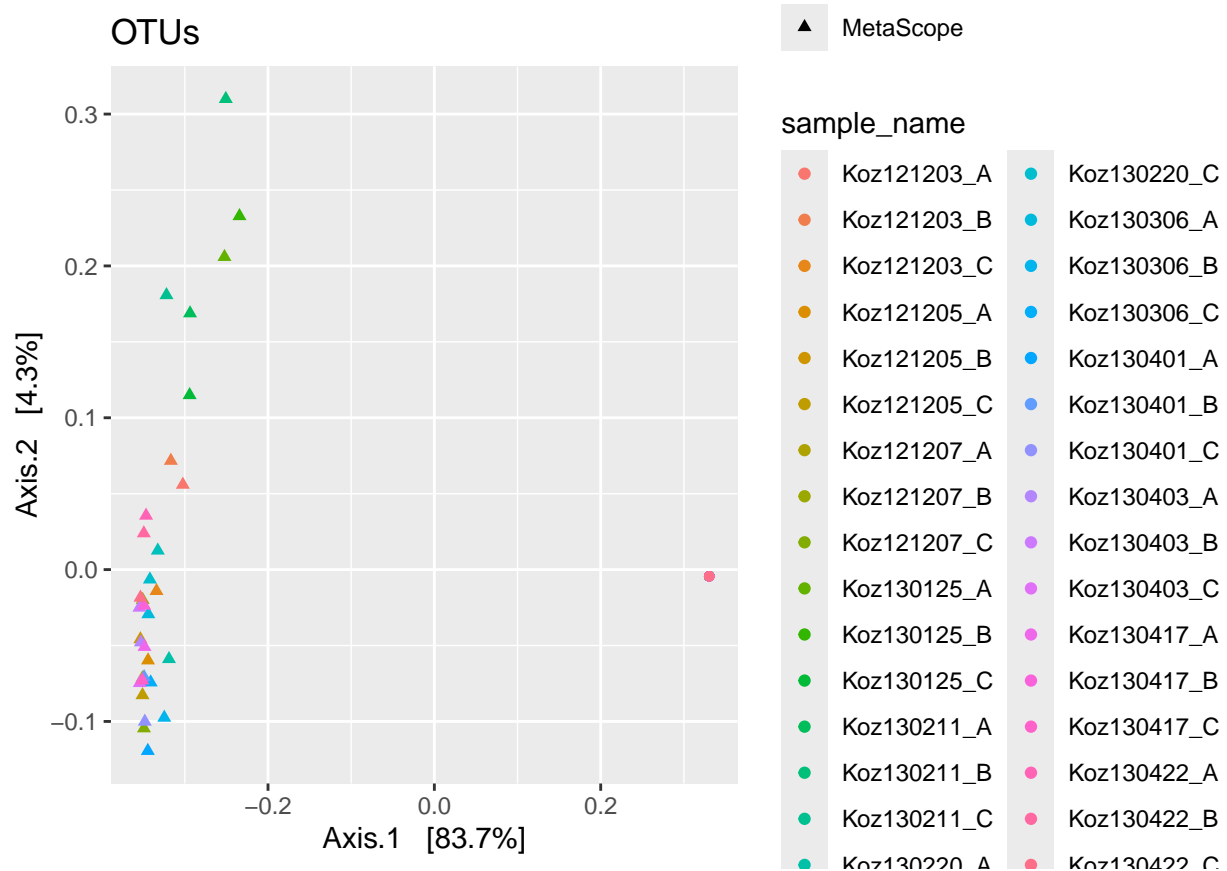
merged_tax_mat <- ms_mats[[2]] |>
  as.data.frame() |>
  bind_rows(ground_truth_tax_mat) |>
  dplyr::select(-kingdom) |>
  distinct(.keep_all = TRUE) |>
  as.matrix()

merged_samples_df <- matrix(c(rep(unique(ground_truth_df$name), times = 2),
  rep(c("MetaScope", "Ground_Truth"), each = 33)),
  nrow = ncol(merged_otu_mat), ncol = 2,
  dimnames = list(colnames(merged_otu_mat),
    c("sample_name", "pipeline"))) |>
  as.data.frame()

merged_ps <- phyloseq::phyloseq(otu_table(merged_otu_mat, taxa_are_rows = TRUE),
  tax_table(merged_tax_mat),
  sample_data(merged_samples_df))

```

```
merged_ps.ord <- ordinate(merged_ps, "PCoA", "jsd")
plot_ordination(merged_ps, merged_ps.ord, type="samples", color="sample_name", shape="pipeline",
                title="OTUs")
```



Using Vegan to determine distance matrices

```
ground_truth_taxonomy_taxids <-
  taxonomizr::getId(ground_truth_taxonomy$species,
                    accessions_path)

# Helper function to generate distance metrics from metascope files
dist_ms_to_ground_truth <- function(file_paths, pipeline, distance_metric) {
  dfs_list <- file_paths |>
  set_names(function(x) sub("\\\\.*$", "", basename(x))) |>
  map(function(x) {
    df <- read.csv(x)
    sample_name <- sub("\\\\.*$", "", basename(x))
    df <- df |>
    select(TaxonomyID, Genome, readsEM) |>
    dplyr::rename(!paste0(sample_name, ".", pipeline) := readsEM)
    return(df)
  })
  dfs_merged <- reduce(dfs_list, full_join, by = c("TaxonomyID", "Genome")) |>
```

```

    mutate(across(where(is.numeric), ~ replace_na(.x, 0))) |>
    dplyr::select(-Genome) |>
    tibble::column_to_rownames("TaxonomyID") |>
    dplyr::mutate_if(is.numeric, ~ . / sum(.))

# Generate Matrix used for vegan::vegdist and full join ground_truth taxonomy
ms_dist_mat <- full_join(dfs_merged |>
  tibble::rownames_to_column("TaxonomyID") |>
  mutate(TaxonomyID = as.numeric(TaxonomyID)),
  data.frame(TaxonomyID = as.numeric(ground_truth_taxonomy_taxids),
    Ground_Truth = rep(1/length(ground_truth_taxonomy_taxids),
      length(ground_truth_taxonomy_taxids))),
  by = "TaxonomyID") |>
  mutate(across(everything(), ~ replace_na(.x, 0))) |>
  column_to_rownames("TaxonomyID") |>
  filter(rowSums(across(where(is.numeric))) > 0) |>
  t()

dist_mat <- vegan::vegdist(ms_dist_mat, method = distance_metric) |> as.matrix() |>
  as.data.frame() |>
  filter(row_number() <= n()-1) |>
  pull(Ground_Truth)

return(dist_mat)
}

## Function to generate distances for all outputs
dist_from_ground_truth <- function(distance_metric) {
  ms_dist <- dist_ms_to_ground_truth(list.files(path = "data_processed/kozich_2013/results",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
    pipeline = "MetaScope",
    distance_metric = distance_metric)
  ms_p_dist <- dist_ms_to_ground_truth(list.files(path = "data_processed/kozich_2013/results_priors",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
    pipeline = "MetaScope Priors",
    distance_metric = distance_metric)
  ms_p_b_dist <- dist_ms_to_ground_truth(list.files(path = "data_processed/kozich_2013/results_metablas",
    pattern = ".metascope_id.csv",
    full.names = TRUE,
    recursive = TRUE),
    pipeline = "MetaBlast",
    distance_metric = distance_metric)

  dada2_dist <- dada2_df |> select(name, taxon, taxid, reads_count) |>
    group_by(taxon, taxid, name) |>
    summarise(reads_count = sum(reads_count), .groups = "drop") |>
    pivot_wider(names_from = name, values_from = reads_count, values_fill = 0) |>

```



```

full_join(data.frame(taxid = as.numeric(ground_truth_taxonomy_taxids),
                    Ground_Truth = rep(1/length(ground_truth_taxonomy_taxids),
                                       length(ground_truth_taxonomy_taxids))),
          by = "taxid") |>
mutate(taxon = if_else(is.na(taxon), taxonomizr::getTaxonomy(taxid, accessions_path, desiredTaxa =
                    taxon)) |>
mutate(across(everything(), ~ replace_na(.x, 0))) |>
tibble::column_to_rownames("taxon") |> # Using taxon for rowname instead of taxid because I labeled
select(-taxid) |>
t() |>
vegan::vegdist(method = distance_metric) |>
as.matrix() |>
as.data.frame() |>
filter(row_number() <= n()-1) |> # Remove the last row which contains ground_truth vs ground_truth
pull(Ground_Truth)

qiime2_dist <- qiime2_df |>
select(name, taxon, taxid, value) |>
group_by(taxon, taxid, name) |>
summarise(reads_count = sum(value), .groups = "drop") |>
pivot_wider(names_from = name, values_from = reads_count, values_fill = 0) |>
full_join(data.frame(taxid = as.numeric(ground_truth_taxonomy_taxids),
                    Ground_Truth = rep(1/length(ground_truth_taxonomy_taxids),
                                       length(ground_truth_taxonomy_taxids))),
          by = "taxid") |>
mutate(taxon = if_else(is.na(taxon), taxonomizr::getTaxonomy(taxid, accessions_path, desiredTaxa =
                    taxon)) |>
mutate(across(everything(), ~ replace_na(.x, 0))) |>
tibble::column_to_rownames("taxon") |> # Using taxon for rowname instead of taxid because I labeled
select(-taxid) |>
t() |>
vegan::vegdist(method = distance_metric) |>
as.matrix() |>
as.data.frame() |>
filter(row_number() <= n()-1) |> # Remove the last row which contains ground_truth vs ground_truth
pull(Ground_Truth)

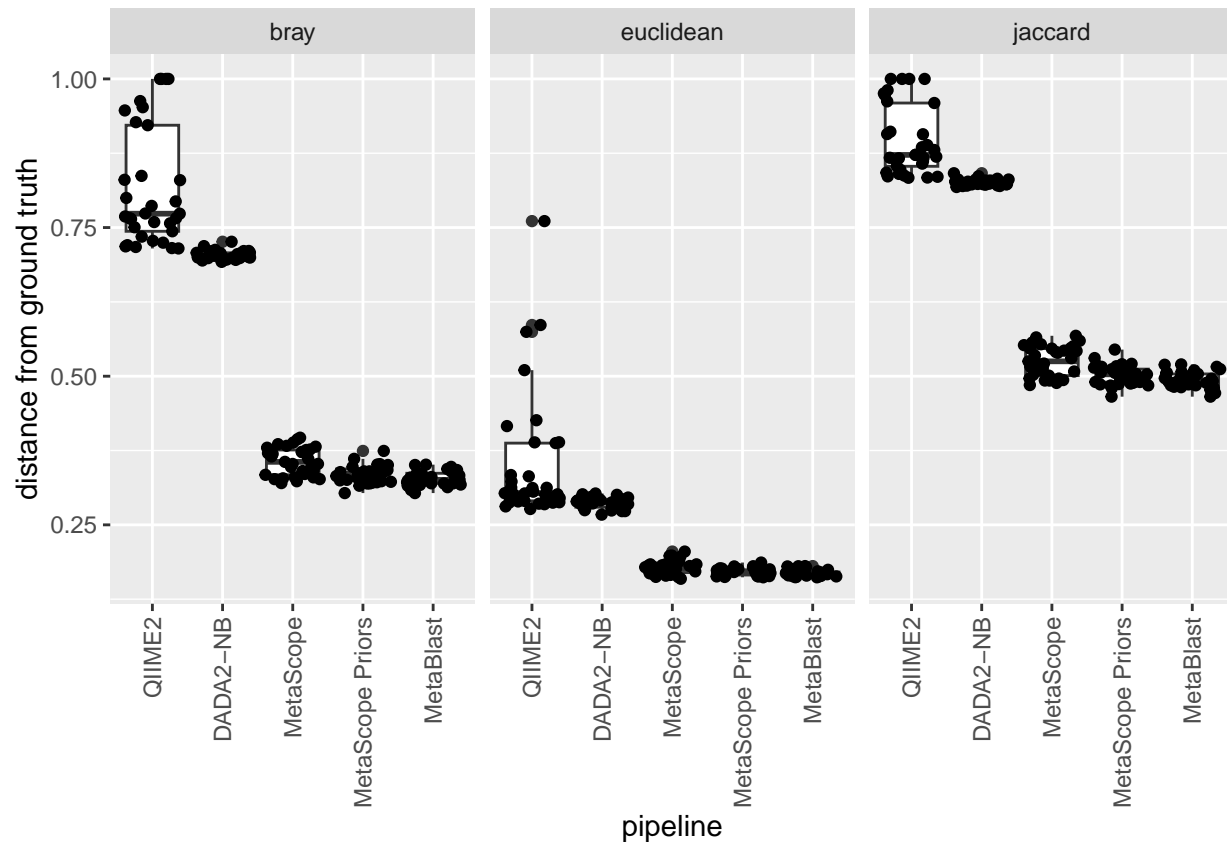
merged_res <- tibble(
  "QIIME2" = qiime2_dist,
  "DADA2-NB" = dada2_dist,
  "MetaScope" = ms_dist,
  "MetaScope Priors" = ms_p_dist,
  "MetaBlast" = ms_p_b_dist,
  "distance_method" = distance_metric) |>
pivot_longer(cols = c("QIIME2", "DADA2-NB", "MetaScope", "MetaScope Priors", "MetaBlast"),
             names_to = "pipeline", values_to = "distance_from_ground_truth")
merged_res$pipeline <- factor(merged_res$pipeline,
                             levels = c("QIIME2", "DADA2-NB", "MetaScope",
                                          "MetaScope Priors", "MetaBlast"))

return(merged_res)
}

distance_res <- map_dfr(c("bray", "jaccard", "euclidean"), dist_from_ground_truth)

```

```
ggplot(distance_res, aes(x = pipeline, y = distance_from_ground_truth)) +
  geom_boxplot() +
  geom_jitter() +
  facet_grid(~distance_method) +
  xlab("pipeline") +
  ylab("distance from ground truth") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
distance_res |>
  group_by(distance_method, pipeline) |>
  summarise(mean = mean(distance_from_ground_truth),
            sd = sd(distance_from_ground_truth))
```

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

```
## # A tibble: 15 x 4
## # Groups:   distance_method [3]
##   distance_method pipeline      mean      sd
##   <chr>          <fct>    <dbl>  <dbl>
## 1 bray          QIIME2    0.818  0.100
## 2 bray          DADA2-NB  0.703  0.00716
## 3 bray          MetaScope 0.357  0.0235
```

```
## 4 bray          MetaScope Priors 0.335 0.0145
## 5 bray          MetaBlast      0.328 0.0117
## 6 euclidean     QIIME2         0.353 0.109
## 7 euclidean     DADA2-NB       0.287 0.00898
## 8 euclidean     MetaScope      0.178 0.0109
## 9 euclidean     MetaScope Priors 0.171 0.00598
## 10 euclidean    MetaBlast      0.169 0.00549
## 11 jaccard       QIIME2         0.897 0.0583
## 12 jaccard       DADA2-NB       0.826 0.00491
## 13 jaccard       MetaScope      0.525 0.0256
## 14 jaccard       MetaScope Priors 0.502 0.0162
## 15 jaccard       MetaBlast      0.494 0.0132
```

Helper function to read MetaScope results

```
read_ms_results <- function(file_paths, pipeline) {
  dfs_list <- file_paths |>
    set_names(function(x) sub("\\..*$", "", basename(x))) |>
    map(function(x) {
      df <- read.csv(x)
      sample_name <- sub("\\..*$", "", basename(x))
      df <- df |>
        select(TaxonomyID, Genome, readsEM) |>
        dplyr::rename(!sample_name := readsEM)
      return(df)
    })
  dfs_merged <- reduce(dfs_list, full_join, by = c("TaxonomyID", "Genome")) |>
    mutate(across(where(is.numeric), ~ replace_na(.x, 0))) |>
    dplyr::select(-Genome) |>
    tibble::column_to_rownames("TaxonomyID") |>
    dplyr::mutate_if(is.numeric, ~ . / sum(.))
  return(dfs_merged)
}
```

Helper Function to calculate precision, recall, and F1 and species level

relative abundance threshold

```
calc_prf <- function(abundance_threshold) {
  dada2_prf <- dada2_df |>
    group_by(name, taxid) |>
    summarise(value = sum(reads_count), .groups = "drop") |> # Merge same sample and taxids together
    filter(value > abundance_threshold) |>
    mutate(abund_weight = 1 - abs(value - 1/21),
           is_tp = taxid %in% ground_truth_taxonomy_taxids) |> # weight by distance from ground truth a
    group_by(name) |>
    summarise(
      TP = sum(is_tp),
      precision = TP / n(),
      recall = TP / length(ground_truth_taxonomy_taxids),
      F1 = 2 * precision * recall / (precision + recall)) |>
    mutate(pipeline = "DADA2-NB")

  qiime2_prf <- qiime2_df |>
    group_by(name, taxid) |>
    summarise(value = sum(value), .groups = "drop") |>
    filter(value > abundance_threshold) |>
```

```

mutate(abund_weight = 1 - abs(value - 1/21), # weight by distance from ground truth abundance
       is_tp = taxid %in% ground_truth_taxonomy_taxids) |>
group_by(name) |>
summarise(
  TP = sum(is_tp * abund_weight),
  precision = TP / n(),
  recall = TP / length(ground_truth_taxonomy_taxids),
  F1 = 2 * precision * recall / (precision + recall)) |>
mutate(pipeline = "QIIME2")

ms_prf <- read_ms_results(list.files(path = "data_processed/kozich_2013/results",
                                   pattern = ".metascope_id.csv",
                                   full.names = TRUE,
                                   recursive = TRUE),
                        pipeline = "MetaScope") |>
tibble::rownames_to_column("taxid") |>
pivot_longer(-taxid, names_to = "name", values_to = "value") |>
group_by(name) |>
filter(value > abundance_threshold) |>
mutate(abund_weight = 1 - abs(value - 1/21),
       is_tp = taxid %in% ground_truth_taxonomy_taxids) |> # weight by distance from ground truth a
summarise(
  TP = sum(is_tp),
  precision = TP / n(),
  recall = TP / length(ground_truth_taxonomy_taxids),
  F1 = 2 * precision * recall / (precision + recall)) |>
mutate(pipeline = "MetaScope")

ms_p_prf <- read_ms_results(list.files(path = "data_processed/kozich_2013/results_priors",
                                   pattern = ".metascope_id.csv",
                                   full.names = TRUE,
                                   recursive = TRUE),
                        pipeline = "MetaScope Priors") |>
tibble::rownames_to_column("taxid") |>
pivot_longer(-taxid, names_to = "name", values_to = "value") |>
group_by(name) |>
filter(value > abundance_threshold) |>
mutate(abund_weight = 1 - abs(value - 1/21),
       is_tp = taxid %in% ground_truth_taxonomy_taxids) |> # weight by distance from ground truth a
summarise(
  TP = sum(is_tp),
  precision = TP / n(),
  recall = TP / length(ground_truth_taxonomy_taxids),
  F1 = 2 * precision * recall / (precision + recall)) |>
mutate(pipeline = "MetaScope Priors")

ms_p_b_prf <- read_ms_results(list.files(path = "data_processed/kozich_2013/results_metablast_priors_",
                                   pattern = ".metascope_id.csv",
                                   full.names = TRUE,
                                   recursive = TRUE),
                        pipeline = "MetaBlast") |>

```

```

tibble::rownames_to_column("taxid") |>
pivot_longer(-taxid, names_to = "name", values_to = "value") |>
group_by(name) |>
filter(value > abundance_threshold) |>
mutate(abund_weight = 1 - abs(value - 1/21),
       is_tp = taxid %in% ground_truth_taxonomy_taxids) |> # weight by distance from ground truth a
summarise(
  TP = sum(is_tp),
  precision = TP / n(),
  recall = TP / length(ground_truth_taxonomy_taxids),
  F1 = 2 * precision * recall / (precision + recall)) |>
mutate(pipeline = "MetaBlast")

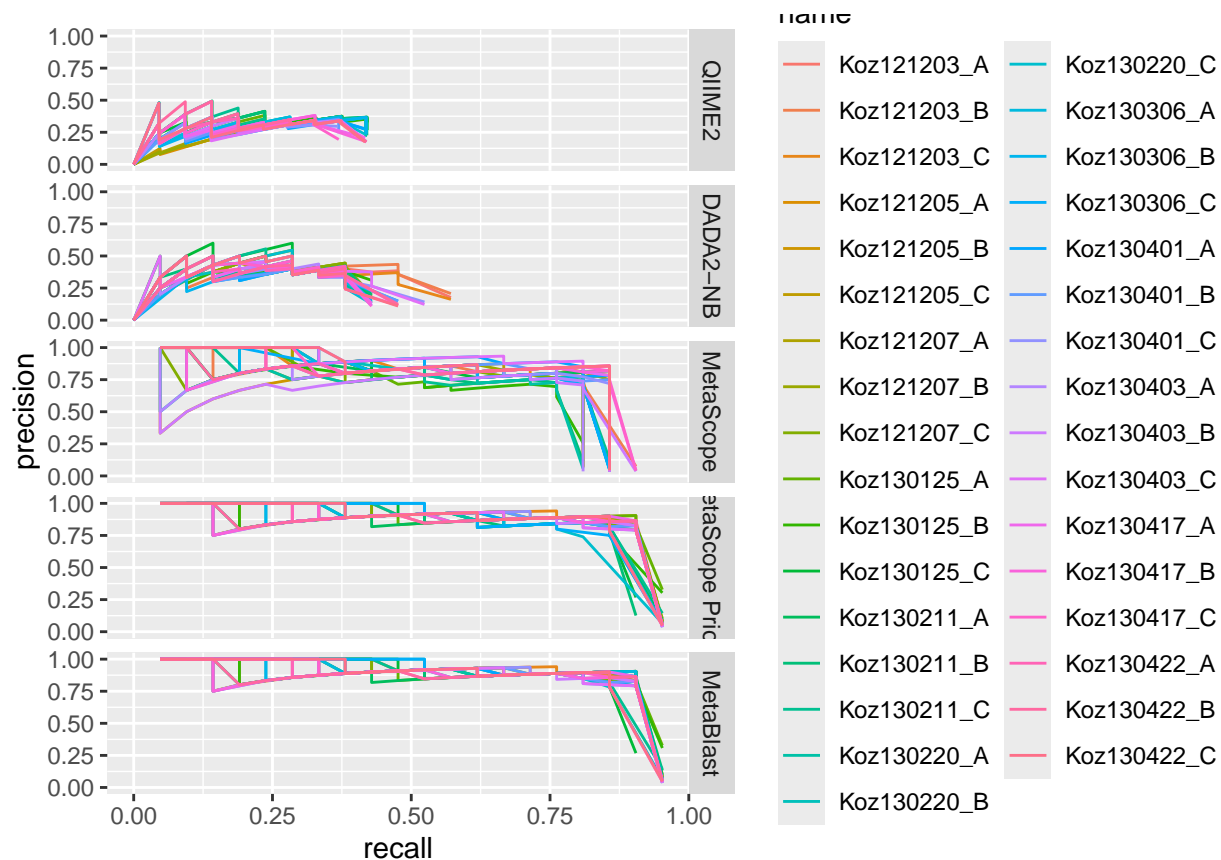
merged_prf <- rbind(dada2_prf, qiime2_prf, ms_prf, ms_p_prf, ms_p_b_prf) |>
#pivot_longer(cols = c(precision, recall, F1), names_to = "metric_scores", values_to = "value") |>
select(-TP) |>
replace_na(list(value = 0))
merged_prf$pipeline <- factor(merged_prf$pipeline,
                             levels = c("QIIME2", "DADA2-NB", "MetaScope",
                                           "MetaScope Priors", "MetaBlast"))

merged_prf$abundance_threshold <- abundance_threshold
return(merged_prf)
}

threshold_vals <- c(0,1e-6,5e-6,1e-5,5e-5,1e-4,5e-4,1e-3,5e-3,1e-1,5e-1, 1)
threshold_vals <- seq(0,0.1, 0.001)
pr_curve_df <- map_dfr(threshold_vals,calc_prf)

# Generate Precision Recall curve
ggplot(pr_curve_df, aes(x = recall, y = precision, color = name)) +
  geom_path() +
  facet_grid(rows = vars(pipeline))

```

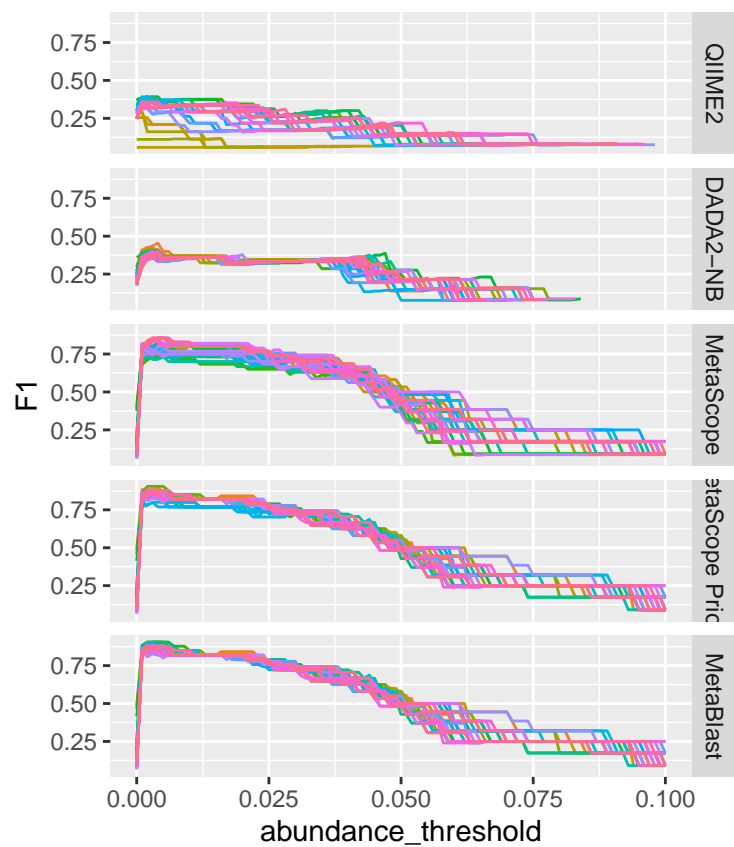


```
# Calculate AUC for precision recall curve
pr_curve_df |>
  group_by(name, pipeline) |>
  arrange(recall) |>
  summarize(cum_area = trapz(recall, precision)) |>
  ungroup() |>
  group_by(pipeline) |>
  summarize(aauc = mean(cum_area))
```

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

```
## # A tibble: 5 x 2
##   pipeline      aauc
##   <fct>        <dbl>
## 1 QIIME2      0.0894
## 2 DADA2-NB    0.154
## 3 MetaScope   0.656
## 4 MetaScope Priors 0.778
## 5 MetaBlast   0.787
```

```
# F1 Curve
ggplot(pr_curve_df, aes(x = abundance_threshold, y = F1, color = name)) +
  geom_path() +
  facet_grid(rows = vars(pipeline)) +
  xlim(0,0.1)
```



name

Koz121203_A	Koz130220_C
Koz121203_B	Koz130306_A
Koz121203_C	Koz130306_B
Koz121205_A	Koz130306_C
Koz121205_B	Koz130401_A
Koz121205_C	Koz130401_B
Koz121207_A	Koz130401_C
Koz121207_B	Koz130403_A
Koz121207_C	Koz130403_B
Koz130125_A	Koz130403_C
Koz130125_B	Koz130417_A
Koz130125_C	Koz130417_B
Koz130211_A	Koz130417_C
Koz130211_B	Koz130422_A
Koz130211_C	Koz130422_B
Koz130220_A	Koz130422_C
Koz130220_B	