

New Microbiome analysis version 2.0

Sri Sai Nandini Ravi

2025-05-24

```
library(agricolae)
library(tidyverse) # for data manipulation and plotting
library(readxl)    # for reading Excel files
library(reshape2)  # for reshaping data
library(ggplot2)   # for plotting
library(phyloseq)
library(dplyr)
library(vegan)
library(ALDEx2)
library(indicspecies)
library(Hmisc)     # For Spearman's correlation
library(igraph)    # For network analysis
library(ggraph)    # For network visualization
library(tidygraph) # For handling network objects
library(dplyr)     # For data wrangling

# Identify sample columns
sample_columns_16s <- grep("^FFAR", colnames(ASV_16s), value = TRUE)
sample_columns_ITS <- grep("^FFAR", colnames(ASV_ITS), value = TRUE)

# Reshape 16S ASV table from wide to long format
asv_16s_long <- ASV_16s %>%
  pivot_longer(cols = all_of(sample_columns_16s),
               names_to = "sample_name",
               values_to = "Abundance") %>%
  mutate(Abundance = as.numeric(Abundance)) # Ensure numeric values

# Reshape ITS ASV table from wide to long format
asv_ITS_long <- ASV_ITS %>%
  pivot_longer(cols = all_of(sample_columns_ITS),
               names_to = "sample_name",
               values_to = "Abundance") %>%
  mutate(Abundance = as.numeric(Abundance)) # Ensure numeric values

# Merge ASV data with metadata
m_16s <- asv_16s_long %>%
  inner_join(MD, by = "sample_name") # Ensure sample names match

m_ITS <- asv_ITS_long %>%
  inner_join(MD, by = "sample_name") # Ensure sample names match
```

```
# Check the first few rows
head(m_16s)
```

```
## # A tibble: 6 x 20
##   OTU_ID      Taxon Phylum Class Order Family Genus Species Confidence Sequence
##   <chr>      <chr> <chr>  <chr> <chr> <chr>  <chr> <chr>      <dbl> <chr>
## 1 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## 2 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## 3 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## 4 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## 5 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## 6 2f9fd738c1f~ d__B~ " p__~ " c_~ " o_~ " f__~ " g_~ " s__u~      0.994 TACGAAG~
## # i 10 more variables: sample_name <chr>, Abundance <dbl>, Project <chr>,
## #   Treatment <chr>, Type <chr>, Sample <chr>, number <dbl>, Day <chr>,
## #   TreatmentDay <chr>, TreatmentType <chr>
```

```
head(m_ITS)
```

```
## # A tibble: 6 x 20
##   OTU_ID      Kingdom Phylum Class Order Family Genus Species Confidence Sequence
##   <chr>      <chr>  <chr>  <chr> <chr> <chr>  <chr> <chr>      <dbl> <chr>
## 1 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## 2 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## 3 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## 4 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## 5 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## 6 1be8c97d3~ k__Fun~ p__As~ c__E~ o__E~ f__As~ g__T~ s__Tal~      0.963 TAAATGC~
## # i 10 more variables: sample_name <chr>, Abundance <dbl>, Project <chr>,
## #   Treatment <chr>, Type <chr>, Sample <chr>, number <dbl>, Day <chr>,
## #   TreatmentDay <chr>, TreatmentType <chr>
```

```
# Convert filtered 16S data to phyloseq format
otu_16s <- m_16s %>%
  select(OTU_ID, sample_name, Abundance) %>%
  pivot_wider(names_from = sample_name, values_from = Abundance, values_fill = 0) %>%
  column_to_rownames("OTU_ID") %>%
  as.matrix() %>%
  otu_table(taxa_are_rows = TRUE)

sample_16s <- sample_data(MD %>% column_to_rownames("sample_name"))

physeq_16s <- phyloseq(otu_16s, sample_16s)

# Calculate Shannon diversity for 16S
ShD_16s <- estimate_richness(physeq_16s, measures = "Shannon") %>%
  rownames_to_column("sample_name") %>%
  inner_join(MD, by = "sample_name")

# Repeat for ITS data
otu_ITS <- m_ITS %>%
  select(OTU_ID, sample_name, Abundance) %>%
  pivot_wider(names_from = sample_name, values_from = Abundance, values_fill = 0) %>%
```

```

column_to_rownames("OTU_ID") %>%
as.matrix() %>%
otu_table(taxa_are_rows = TRUE)

sample_ITS <- sample_data(MD %>% column_to_rownames("sample_name"))

physeq_ITS <- phyloseq(otu_ITS, sample_ITS)

# Calculate Shannon diversity for ITS
ShD_ITS <- estimate_richness(physeq_ITS, measures = "Shannon") %>%
  rownames_to_column("sample_name") %>%
  inner_join(MD, by = "sample_name")

# View results
head(ShD_16s)

```

```

##   sample_name Shannon   Project Treatment Type      Sample number   Day
## 1   FFAR133 6.002217 Greenhouse    ME root _+M+E_21_root      21 day60
## 2   FFAR134 5.088291 Greenhouse    ME root _+M+E_25_root      25 day60
## 3   FFAR135 5.288670 Greenhouse    ME root _+M+E_30_root      30 day60
## 4   FFAR136 5.767925 Greenhouse    ME root _+M+E_11_root      11 day50
## 5   FFAR137 6.012183 Greenhouse    ME root _+M+E_1_root       1 day40
## 6   FFAR138 5.652901 Greenhouse    ME root _+M+E_15_root      15 day50
##   TreatmentDay TreatmentType
## 1   _+M+E_day60    _+M+E_root
## 2   _+M+E_day60    _+M+E_root
## 3   _+M+E_day60    _+M+E_root
## 4   _+M+E_day50    _+M+E_root
## 5   _+M+E_day40    _+M+E_root
## 6   _+M+E_day50    _+M+E_root

```

```
head(ShD_ITS)
```

```

##   sample_name Shannon   Project Treatment Type      Sample number   Day
## 1   FFAR133 2.493577 Greenhouse    ME root _+M+E_21_root      21 day60
## 2   FFAR134 2.922580 Greenhouse    ME root _+M+E_25_root      25 day60
## 3   FFAR135 3.146986 Greenhouse    ME root _+M+E_30_root      30 day60
## 4   FFAR136 2.260911 Greenhouse    ME root _+M+E_11_root      11 day50
## 5   FFAR137 2.828198 Greenhouse    ME root _+M+E_1_root       1 day40
## 6   FFAR138 2.629085 Greenhouse    ME root _+M+E_15_root      15 day50
##   TreatmentDay TreatmentType
## 1   _+M+E_day60    _+M+E_root
## 2   _+M+E_day60    _+M+E_root
## 3   _+M+E_day60    _+M+E_root
## 4   _+M+E_day50    _+M+E_root
## 5   _+M+E_day40    _+M+E_root
## 6   _+M+E_day50    _+M+E_root

```

here we try to see the same significance using box plots.

```

# Load libraries if not loaded
library(dplyr)
library(ggplot2)
library(multcompView)
library(tidyr)

# Step 1: Add Group column and separate Type and Treatment
ShD_16s <- ShD_16s %>%
  mutate(Group = paste(Type, Treatment, sep = "_"))
# Force desired order of x-axis categories
ShD_16s$Type <- factor(ShD_16s$Type, levels = c("root", "rhizo", "soil"))

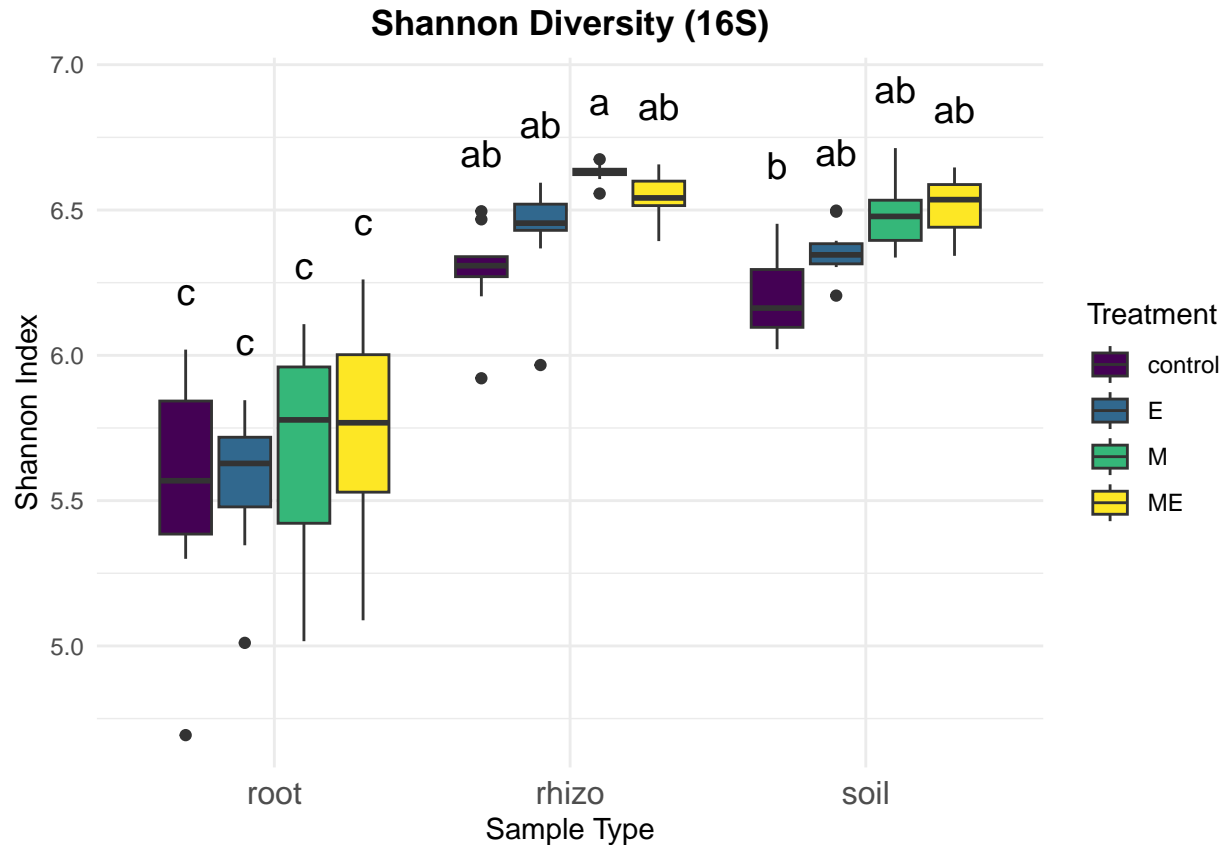
# Step 2: ANOVA and Tukey's HSD test
model_box <- aov(Shannon ~ Group, data = ShD_16s)
tukey_box <- TukeyHSD(model_box)
letters_box <- multcompLetters(tukey_box$Group[, "p adj"])$Letters
letter_df_box <- data.frame(Group = names(letters_box), Letters = letters_box)

# Step 3: Separate Group into Type and Treatment
letter_df_box <- letter_df_box %>%
  separate(Group, into = c("Type", "Treatment"), sep = "_") %>%
  mutate(Type = factor(Type, levels = c("root", "rhizo", "soil")),
         Treatment = factor(Treatment, levels = c("control", "E", "M", "ME")))

# Step 4: Merge with original data to get position for each box
label_positions <- ShD_16s %>%
  group_by(Type, Treatment) %>%
  summarise(y_pos = max(Shannon) + 0.2, .groups = "drop") %>%
  left_join(letter_df_box, by = c("Type", "Treatment"))

# Step 5: Create the grouped boxplot
ggplot(ShD_16s, aes(x = Type, y = Shannon, fill = Treatment)) +
  geom_boxplot(position = position_dodge(0.8), width = 0.7) +
  geom_text(data = label_positions,
            aes(x = Type, y = y_pos, group = Treatment, label = Letters),
            position = position_dodge(0.8), size = 5) +
  scale_fill_viridis_d() +
  labs(title = "Shannon Diversity (16S)", x = "Sample Type", y = "Shannon Index") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 12),
        plot.title = element_text(hjust = 0.5, face = "bold"))

```



Now we are trying to make a box plot for ITS using the same technique as 16s

```
# Load required libraries
library(dplyr)
library(ggplot2)
library(multcompView)
library(tidyr)

# Step 1: Add Group column for ITS
ShD_ITS <- ShD_ITS %>%
  mutate(Group = paste(Type, Treatment, sep = "_"))

# Force desired order of x-axis (root → rhizo → soil)
ShD_ITS$Type <- factor(ShD_ITS$Type, levels = c("root", "rhizo", "soil"))

# Step 2: Run ANOVA and Tukey HSD
model_ITS <- aov(Shannon ~ Group, data = ShD_ITS)
tukey_ITS <- TukeyHSD(model_ITS)
letters_ITS <- multcompLetters(tukey_ITS$Group[, "p adj"])$Letters
letter_df_ITS <- data.frame(Group = names(letters_ITS), Letters = letters_ITS)

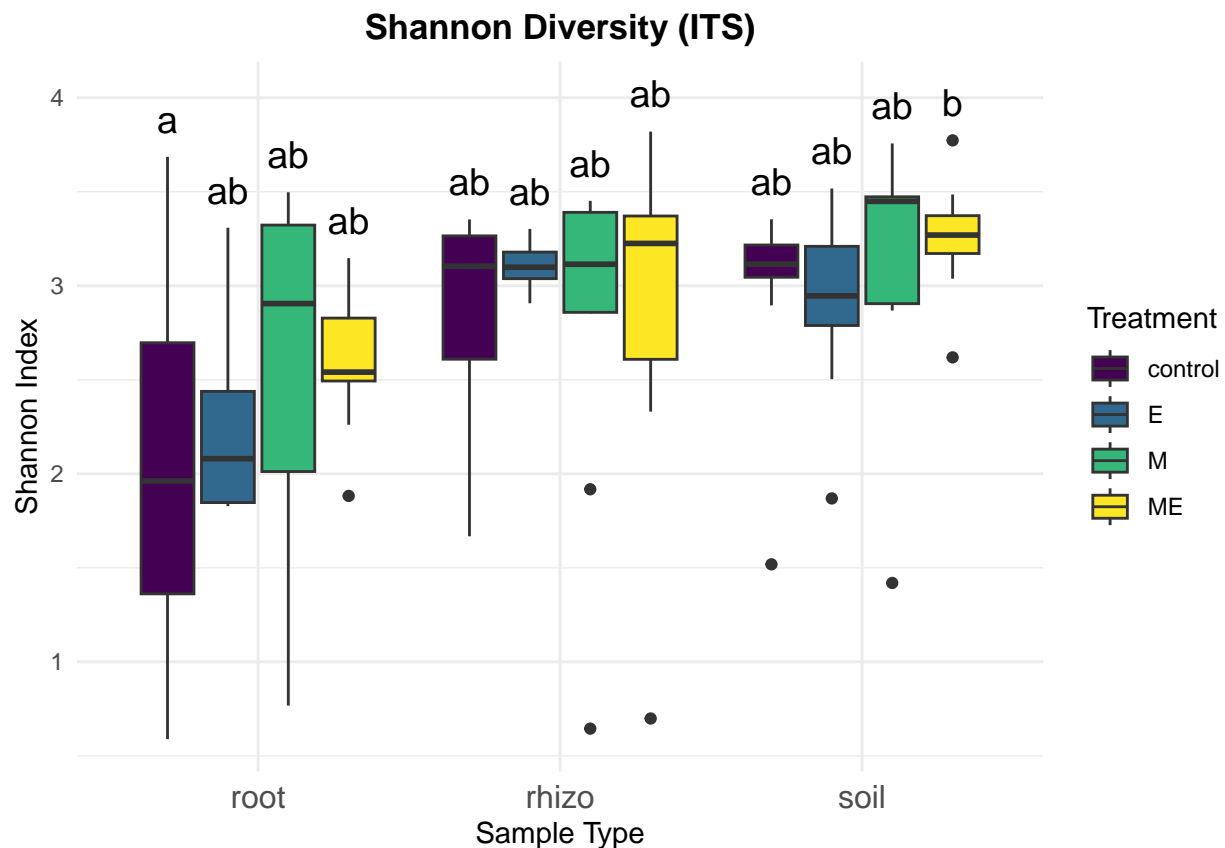
# Step 3: Separate Group into Type and Treatment
letter_df_ITS <- letter_df_ITS %>%
  separate(Group, into = c("Type", "Treatment"), sep = "_") %>%
  mutate(Type = factor(Type, levels = c("root", "rhizo", "soil")),
         Treatment = factor(Treatment, levels = c("control", "E", "M", "ME")))
```

```

# Step 4: Get y-position for each label
label_positions_ITS <- ShD_ITS %>%
  group_by(Type, Treatment) %>%
  summarise(y_pos = max(Shannon) + 0.2, .groups = "drop") %>%
  left_join(letter_df_ITS, by = c("Type", "Treatment"))

# Step 5: Plot ITS grouped boxplot with letters
ggplot(ShD_ITS, aes(x = Type, y = Shannon, fill = Treatment)) +
  geom_boxplot(position = position_dodge(0.8), width = 0.7) +
  geom_text(data = label_positions_ITS,
    aes(x = Type, y = y_pos, group = Treatment, label = Letters),
    position = position_dodge(0.8), size = 5) +
  scale_fill_viridis_d() +
  labs(title = "Shannon Diversity (ITS)", x = "Sample Type", y = "Shannon Index") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, face = "bold"))

```



Beta Diversity, Here we make PCoA and also we show differences between M and ME

```

# Load libraries if not already
library(phyloseq)
library(ggplot2)
library(vegan)
library(dplyr)

```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
# -- 16S Ordination --
```

```
bray_16s <- phyloseq::distance(physeq_16s, method = "bray")  
pcoa_16s <- ordinate(physeq_16s, method = "PCoA", distance = bray_16s)
```

```
# Filter and run PERMANOVA for 16S
```

```
MD_16s_filtered <- MD %>%  
  filter(sample_name %in% rownames(as.matrix(bray_16s))) %>%  
  drop_na(Type, Treatment)
```

```
permanova_16s <- adonis2(bray_16s ~ Treatment, data = MD_16s_filtered, permutations = 999)  
r2_16s <- round(permanova_16s$R2[1], 3)  
pval_16s <- permanova_16s$`Pr(>F)`[1]
```

```
# PCoA plot for 16S
```

```
plot_16s <- plot_ordination(physeq_16s, pcoa_16s, color = "Treatment", shape = "Type") +  
  geom_point(size = 4, alpha = 0.8) +  
  stat_ellipse(type = "t", linetype = "dashed") +  
  scale_color_viridis_d() +  
  theme_minimal() +  
  labs(  
    title = "PCoA (16S) - Bray-Curtis",  
    subtitle = paste("PERMANOVA: R2 =", r2_16s, ", p =", pval_16s),  
    x = "PCoA Axis 1",  
    y = "PCoA Axis 2"  
  )
```

```
# -- ITS Ordination --
```

```
bray_ITS <- phyloseq::distance(physeq_ITS, method = "bray")  
pcoa_ITS <- ordinate(physeq_ITS, method = "PCoA", distance = bray_ITS)
```

```
# Filter and run PERMANOVA for ITS
```

```
MD_ITS_filtered <- MD %>%  
  filter(sample_name %in% rownames(as.matrix(bray_ITS))) %>%  
  drop_na(Type, Treatment)
```

```
permanova_ITS <- adonis2(bray_ITS ~ Treatment, data = MD_ITS_filtered, permutations = 999)  
r2_ITS <- round(permanova_ITS$R2[1], 3)  
pval_ITS <- permanova_ITS$`Pr(>F)`[1]
```

```
# PCoA plot for ITS
```

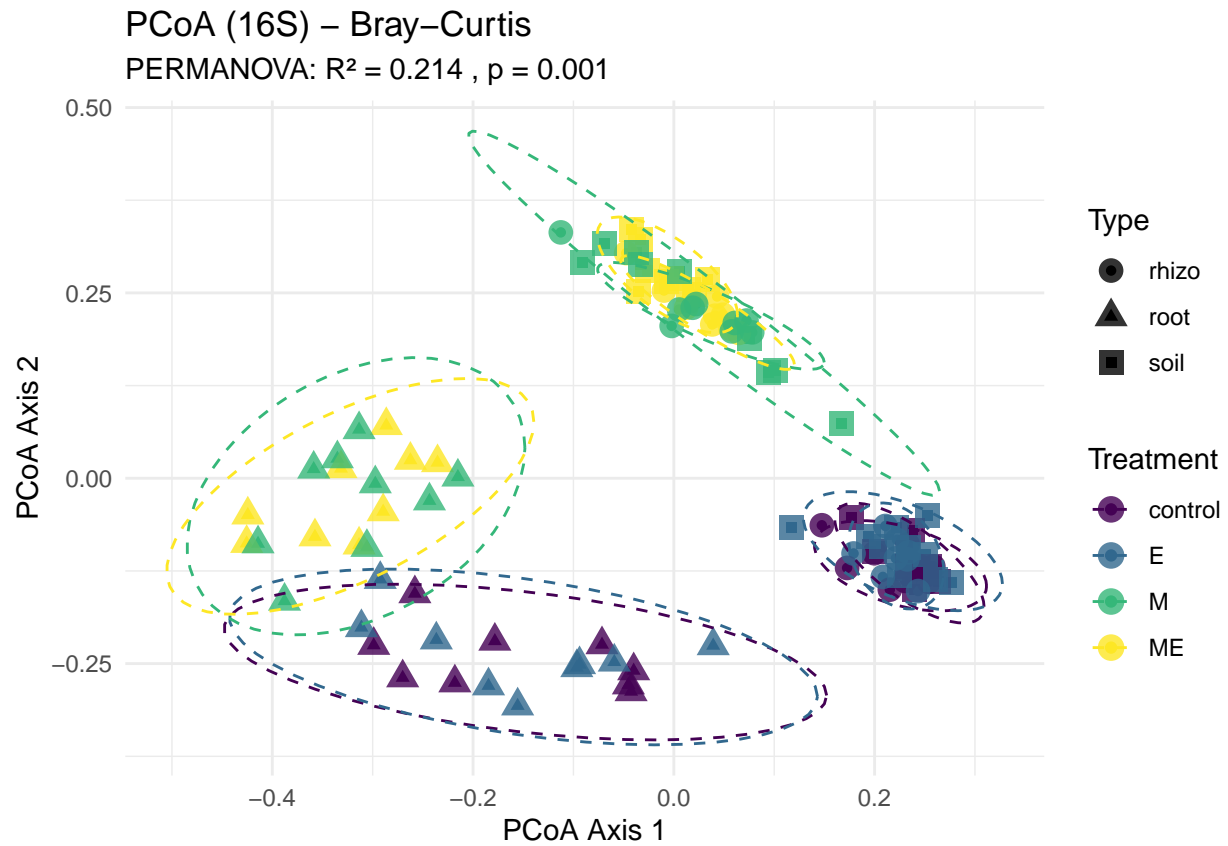
```
plot_ITS <- plot_ordination(physeq_ITS, pcoa_ITS, color = "Treatment", shape = "Type") +  
  geom_point(size = 4, alpha = 0.8) +  
  stat_ellipse(type = "t", linetype = "dashed") +  
  scale_color_viridis_d() +  
  theme_minimal() +  
  labs(  
    title = "PCoA (ITS) - Bray-Curtis",  
    subtitle = paste("PERMANOVA: R2 =", r2_ITS, ", p =", pval_ITS),
```

```

x = "PCoA Axis 1",
y = "PCoA Axis 2"
)

# Display both plots
plot_16s

```

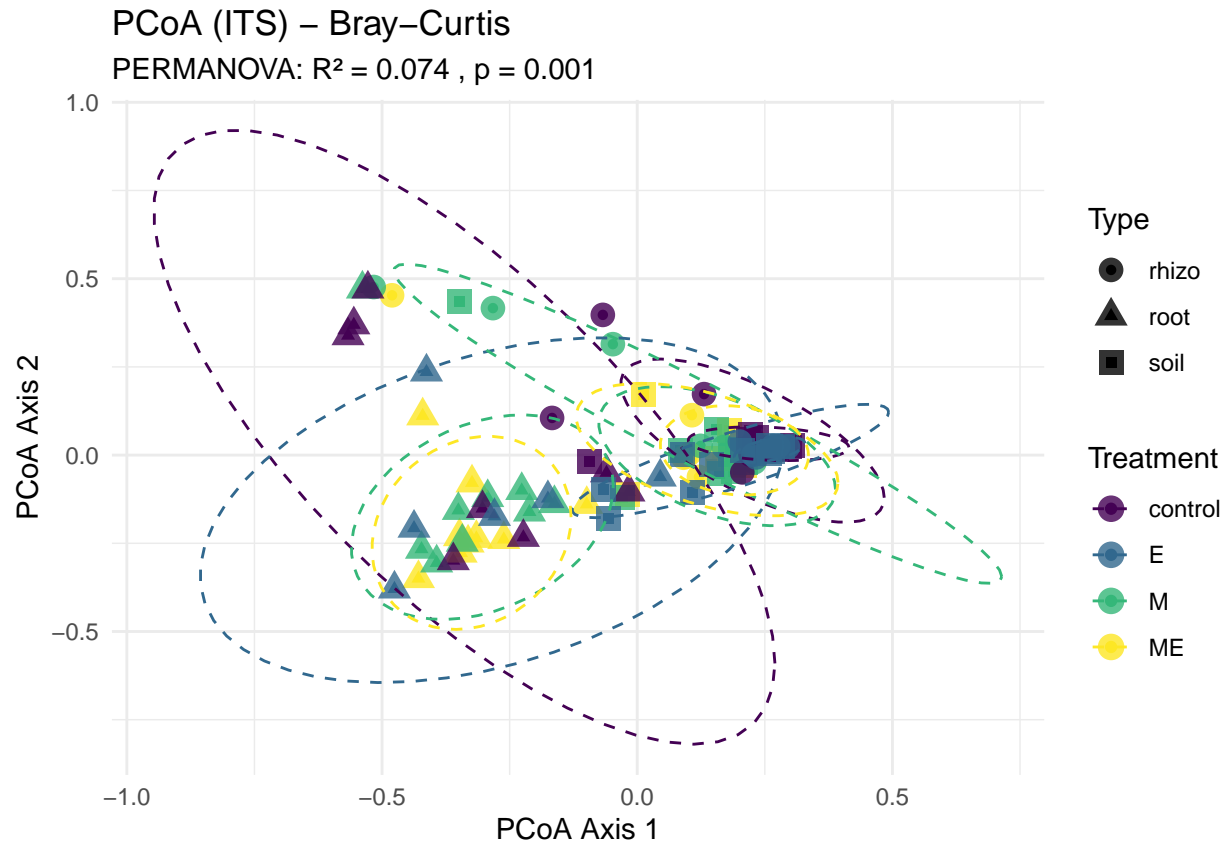


```
plot_ITS
```

```

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

```

```
# Load libraries (if not already loaded)
library(phyloseq)
library(ggplot2)
library(vegan)
library(dplyr)
library(viridis)
library(tibble)

# 16S Bray-Curtis distance and ordination
bray_16s <- phyloseq::distance(physeq_16s, method = "bray")
pcoa_16s <- ordinate(physeq_16s, method = "PCoA", distance = bray_16s)

# Filter metadata to match distance matrix
MD_16s_filtered <- MD %>%
  filter(sample_name %in% rownames(as.matrix(bray_16s))) %>%
  drop_na(Type, Treatment)

# Fix: Run PERMANOVA without Type
permanova_16s <- adonis2(bray_16s ~ Treatment, data = MD_16s_filtered, permutations = 999)

# Extract R² and p-value
r2_16s <- round(permanova_16s$R2[1], 3) # [1] because now Treatment is first
pval_16s <- permanova_16s$`Pr(>F)`[1]

# Create ordination dataframe and join metadata
ord_df_16s <- plot_ordination(physeq_16s, pcoa_16s, justDF = TRUE) %>%
```

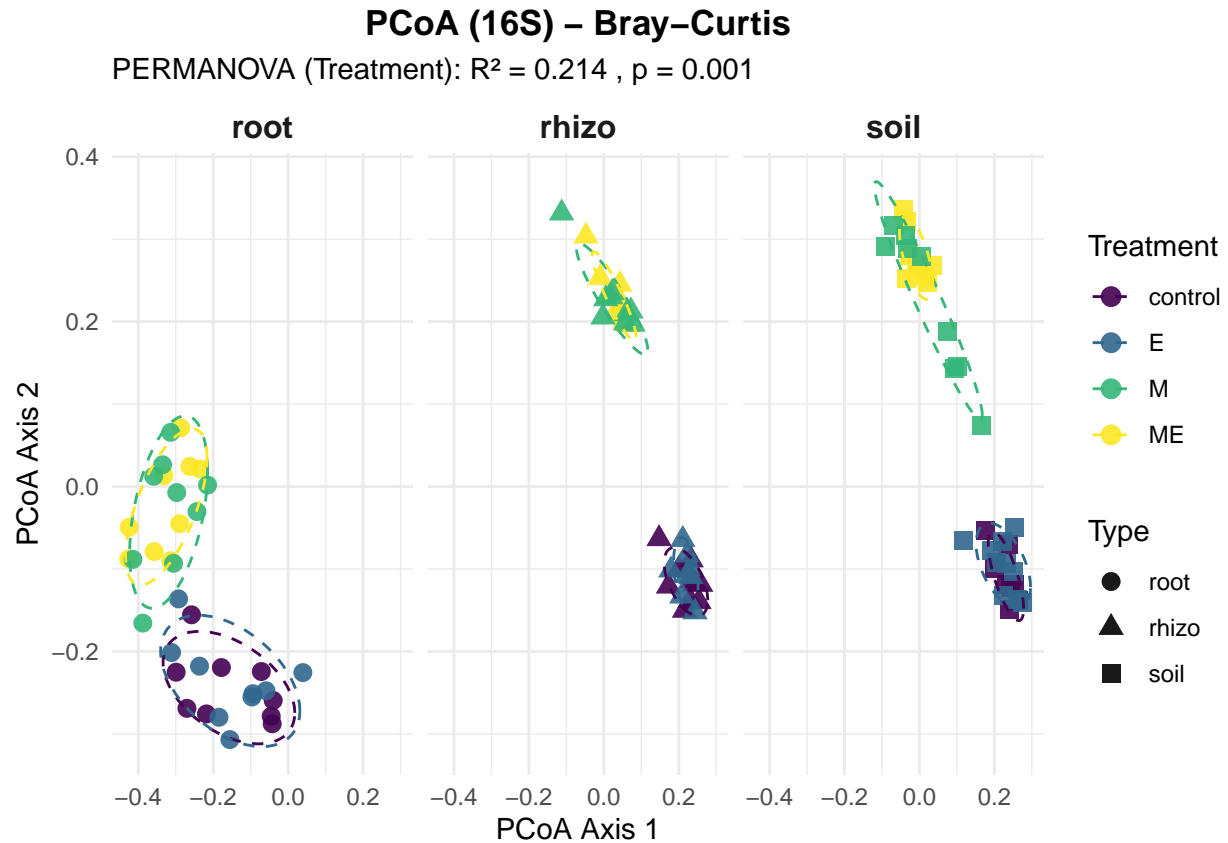
```

tibble::rownames_to_column("sample_name") %>%
left_join(MD, by = "sample_name") %>%
rename(
  Treatment = Treatment.x,
  Type = Type.x
)

# Reorder facet levels: root → rhizo → soil
ord_df_16s$Type <- factor(ord_df_16s$Type, levels = c("root", "rhizo", "soil"))

# -- Create faceted PCoA plot with ellipses --
ggplot(ord_df_16s, aes(x = Axis.1, y = Axis.2, color = Treatment)) +
  geom_point(aes(shape = Type), size = 3, alpha = 0.9) +
  stat_ellipse(type = "norm", level = 0.68, linetype = "dashed") +
  facet_wrap(~Type) +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(
    title = "PCoA (16S) - Bray-Curtis",
    subtitle = paste("PERMANOVA (Treatment): R2 =", r2_16s, ", p =", pval_16s),
    x = "PCoA Axis 1",
    y = "PCoA Axis 2"
  ) +
  theme(
    strip.text = element_text(face = "bold", size = 12),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "right"
  )

```



```
# Load required libraries
library(phyloseq)
library(ggplot2)
library(vegan)
library(dplyr)
library(viridis)
library(tibble)

# ITS Bray-Curtis distance and ordination
bray_ITS <- phyloseq::distance(physeq_ITS, method = "bray")
pcoa_ITS <- ordinate(physeq_ITS, method = "PCoA", distance = bray_ITS)

# filter metadata to match distance matrix and avoid NA issues
MD_ITS_filtered <- MD %>%
  filter(sample_name %in% rownames(as.matrix(bray_ITS))) %>%
  drop_na(Type, Treatment)

# Run PERMANOVA on Treatment only
permanova_ITS <- adonis2(bray_ITS ~ Treatment, data = MD_ITS_filtered, permutations = 999)
r2_ITS <- round(permanova_ITS$R2[1], 3)
pval_ITS <- permanova_ITS$`Pr(>F)`[1]

# Create ordination dataframe and join metadata
ord_df_ITS <- plot_ordination(physeq_ITS, pcoa_ITS, justDF = TRUE) %>%
  tibble::rownames_to_column("sample_name") %>%
  left_join(MD, by = "sample_name") %>%
```

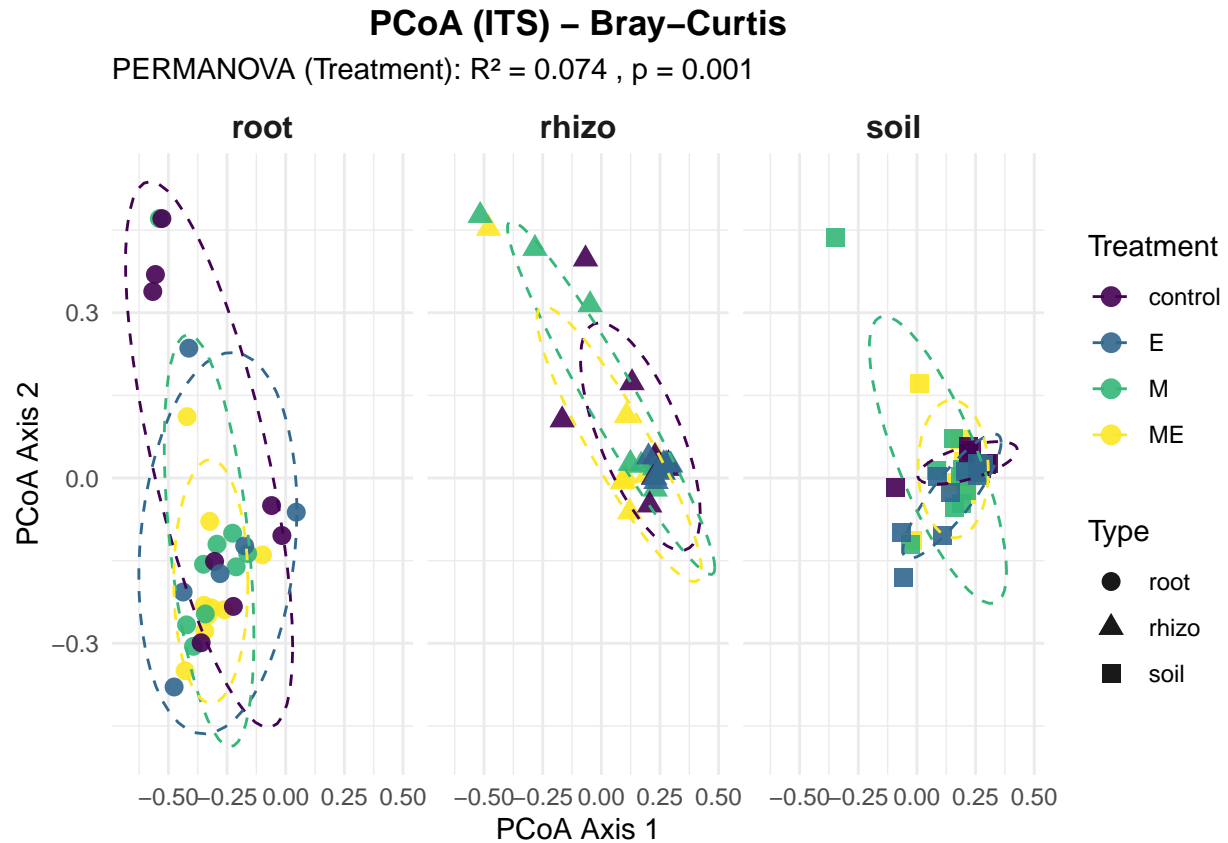
```

rename(
  Treatment = Treatment.x,
  Type = Type.x
)

# Reorder facet levels: root → rhizo → soil
ord_df_ITS$Type <- factor(ord_df_ITS$Type, levels = c("root", "rhizo", "soil"))

# Create faceted PCoA plot with ellipses
ggplot(ord_df_ITS, aes(x = Axis.1, y = Axis.2, color = Treatment)) +
  geom_point(aes(shape = Type), size = 3, alpha = 0.9) +
  stat_ellipse(type = "norm", level = 0.68, linetype = "dashed") +
  facet_wrap(~Type) +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(
    title = "PCoA (ITS) - Bray-Curtis",
    subtitle = paste("PERMANOVA (Treatment): R2 =", r2_ITS, ", p =", pval_ITS),
    x = "PCoA Axis 1",
    y = "PCoA Axis 2"
  ) +
  theme(
    strip.text = element_text(face = "bold", size = 12),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "right"
  )

```



Indicator species analysi

```
# Load required libraries
library(tidyverse)
library(indicspecies)
library(vegan)

# STEP 1: Subset metadata
Meta_root <- MD %>% filter(Type == "root")
Meta_rhizo <- MD %>% filter(Type == "rhizo")
Meta_soil <- MD %>% filter(Type == "soil")

# STEP 2: Function to prep ASV matrix (fixed)
prepare_matrix <- function(ASV, metadata) {
  sample_cols <- metadata$sample_name

  # Fix: Move OTU_ID outside taxonomy_cols, match your column order
  taxonomy_cols <- c("Taxon", "Phylum", "Class", "Order", "Family", "Genus", "Species")

  filtered <- ASV[, c("OTU_ID", taxonomy_cols, sample_cols)]

  mat <- filtered %>%
    select(-all_of(taxonomy_cols)) %>%
    column_to_rownames("OTU_ID") %>%
    t() %>%
    as.data.frame()
```

```

  list(matrix = mat, taxonomy = filtered[, c("OTU_ID", taxonomy_cols)])
}

# STEP 3: Prepare matrices
asv_root <- prepare_matrix(ASV_16s, Meta_root)
asv_rhizo <- prepare_matrix(ASV_16s, Meta_rhizo)
asv_soil <- prepare_matrix(ASV_16s, Meta_soil)

# STEP 4: Run ISA
run_isa <- function(matrix, meta) {
  group <- meta$Treatment[match(rownames(matrix), meta$sample_name)]
  multipatt(matrix, group, func = "r.g", control = how(nperm = 999))
}

isa_root <- run_isa(asv_root$matrix, Meta_root)
isa_rhizo <- run_isa(asv_rhizo$matrix, Meta_rhizo)
isa_soil <- run_isa(asv_soil$matrix, Meta_soil)

# --- STEP 5: Extract significant species ---
get_sig_species <- function(isa_result, taxonomy, type) {
  as.data.frame(isa_result$sign) %>%
    rownames_to_column("OTU_ID") %>%
    filter(p.value <= 0.05) %>%
    left_join(taxonomy, by = "OTU_ID") %>%
    mutate(SampleType = type)
}

sig_root <- get_sig_species(isa_root, asv_root$taxonomy, "Root")
sig_rhizo <- get_sig_species(isa_rhizo, asv_rhizo$taxonomy, "Rhizo")
sig_soil <- get_sig_species(isa_soil, asv_soil$taxonomy, "Soil")
sig_all <- bind_rows(sig_root, sig_rhizo, sig_soil)

# Clean + Rename All Variables ---
isa_results_clean <- sig_all %>%
  filter(
    !is.na(Species),
    !grepl("uncultured|metagenome|metagenomic|environmental", Species, ignore.case = TRUE),
    Species != ""
  ) %>%
  mutate(
    Treatment = case_when(
      s.control == 1 ~ "Control",
      s.E == 1 ~ "E",
      s.M == 1 ~ "M",
      s.ME == 1 ~ "ME"
    )
  )

# --- Define calc_rel_abund function ---
calc_rel_abund <- function(asv_matrix, taxonomy_df, metadata_df, sample_type) {
  rel_abund_df <- asv_matrix %>%
    mutate(across(everything(), ~ .x / sum(.x))) %>%
    rownames_to_column("Sample") %>%
    pivot_longer(-Sample, names_to = "OTU_ID", values_to = "rel_abundance") %>%

```

```

    left_join(taxonomy_df, by = "OTU_ID") %>%
    left_join(metadata_df %>% select(sample_name, Treatment), by = c("Sample" = "sample_name")) %>%
    group_by(Species, Treatment) %>%
    summarise(rel_abundance = mean(rel_abundance, na.rm = TRUE), .groups = "drop") %>%
    mutate(SampleType = sample_type)

  return(rel_abund_df)
}

# Step 1: Calculate rel. abundance per sample type
abund_root <- calc_rel_abund(asv_root$matrix, asv_root$taxonomy, Meta_root, "Root")
abund_rhizo <- calc_rel_abund(asv_rhizo$matrix, asv_rhizo$taxonomy, Meta_rhizo, "Rhizo")
abund_soil <- calc_rel_abund(asv_soil$matrix, asv_soil$taxonomy, Meta_soil, "Soil")

# Step 2: Combine into one master abundance table
abund_all <- bind_rows(abund_root, abund_rhizo, abund_soil)

# --- Merge with relative abundance ---
isa_combined <- isa_results_clean %>%
  left_join(abund_all, by = c("Species", "Treatment", "SampleType"))

# --- Summarize to remove duplicates
isa_summarized <- isa_combined %>%
  group_by(Species, SampleType, Treatment) %>%
  summarise(MeanAbundance = mean(rel_abundance, na.rm = TRUE), .groups = "drop")

# --- Select top 20 by total IndVal
top20_clean_species <- isa_results_clean %>%
  group_by(Species) %>%
  summarise(TotalStat = sum(stat, na.rm = TRUE), .groups = "drop") %>%
  arrange(desc(TotalStat)) %>%
  slice_head(n = 20) %>%
  pull(Species)

# --- Filter for top 20
isa_top20 <- isa_summarized %>%
  filter(Species %in% top20_clean_species)

# --- Pivot into clean wide format
isa_top20_table <- isa_top20 %>%
  mutate(Treatment = factor(Treatment, levels = c("Control", "M", "E", "ME"))) %>%
  pivot_wider(
    names_from = c(SampleType, Treatment),
    values_from = MeanAbundance,
    values_fn = mean
  ) %>%
  arrange(Species)

isa_top20_table_percent <- isa_top20_table %>%
  mutate(across(where(is.numeric), ~ round(.x * 100, 2)))

write.csv(isa_top20_table_percent, "ISA_16s_Top20_RA_in_%.csv", row.names = FALSE)

```

ISA plot for 16s

```
library(ggplot2)
library(viridis)
library(tidyverse)

# --- Merge p-values and indicator values with summary data ---
isa_plot_data <- isa_combined %>%
  filter(Species %in% top20_clean_species) %>%
  group_by(Species, SampleType, Treatment) %>%
  summarise(
    MeanAbundance = mean(rel_abundance, na.rm = TRUE),
    P_value = mean(p.value, na.rm = TRUE),
    IndVal = mean(stat, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(Species_IndVal = paste0(Species, " (", round(IndVal, 2), ")"))

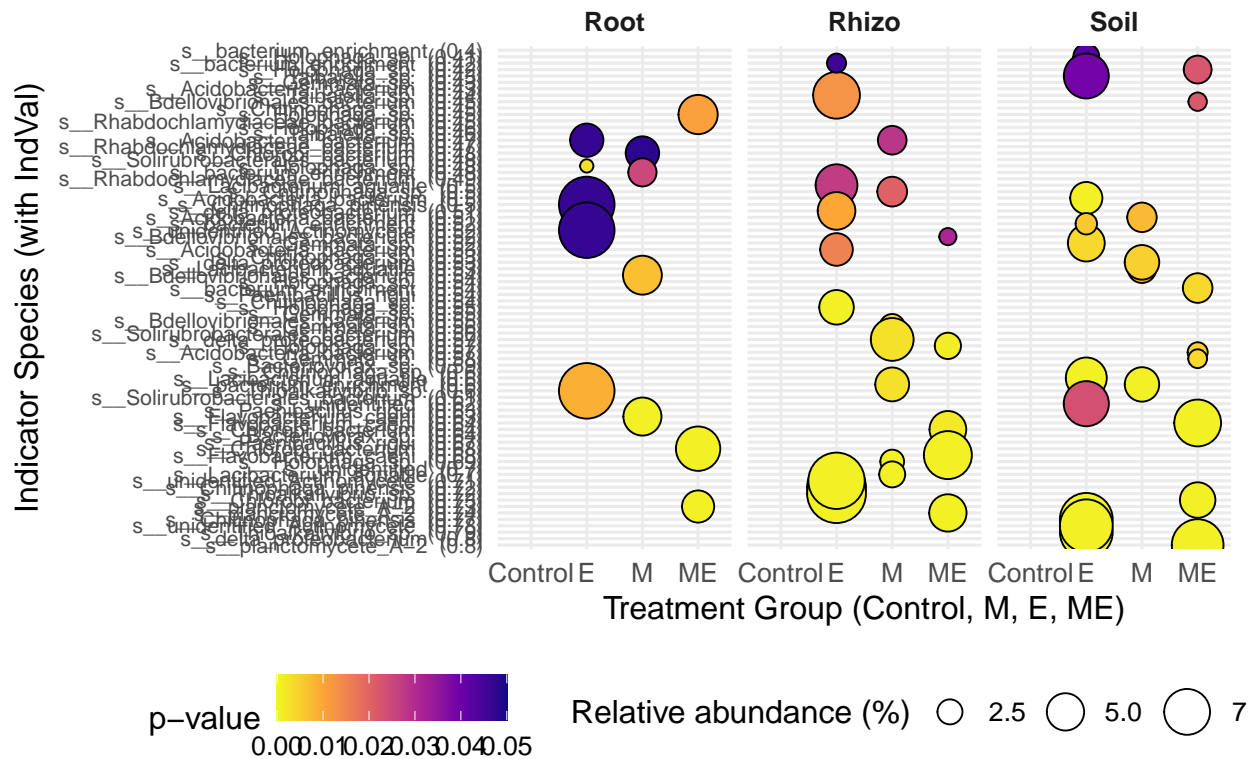
# --- Order species by IndVal descending ---
isa_plot_data$Species_IndVal <- factor(isa_plot_data$Species_IndVal,
  levels = isa_plot_data %>%
    group_by(Species_IndVal) %>%
    summarise(mean_IndVal = mean(IndVal)) %>%
    arrange(desc(mean_IndVal)) %>%
    pull(Species_IndVal))

isa_plot_data <- isa_plot_data %>%
  mutate(
    SampleType = factor(SampleType, levels = c("Root", "Rhizo", "Soil")), # 1. Correct facet order
    Treatment = factor(Treatment, levels = c("Control", "E", "M", "ME")), # (optional if needed)
    MeanAbundance = MeanAbundance * 100 # . Scale abundance to %
  )

# --- Plot ---
ggplot(isa_plot_data, aes(x = Treatment, y = Species_IndVal)) +
  geom_point(aes(size = MeanAbundance, fill = P_value), shape = 21, color = "black") +
  facet_wrap(~SampleType, scales = "free_x") +
  scale_size(range = c(1, 10), name = "Relative abundance (%)") +
  scale_fill_viridis_c(option = "plasma", direction = -1, name = "p-value", limits = c(0, 0.05)) +
  theme_minimal(base_size = 12) +
  labs(
    title = "Indicator Species Plot (16S)",
    x = "Treatment Group (Control, M, E, ME)",
    y = "Indicator Species (with IndVal)"
  ) +
  theme(
    axis.text.y = element_text(size = 8),
    legend.position = "bottom",
    strip.text = element_text(face = "bold") # optional, for bold facet labels
  )
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## ('geom_point()').
```


Indicator Species Plot (16S)



Now for ITS

```
# --- STEP 1: Subset metadata by sample type ---
MetaITS_root <- MD %>% filter(Type == "root")
MetaITS_rhizo <- MD %>% filter(Type == "rhizo")
MetaITS_soil <- MD %>% filter(Type == "soil")

# --- STEP 2: Function to prepare matrix ---
prepare_its_matrix <- function(ASV, metadata) {
  sample_cols <- metadata$sample_name
  taxonomy_cols <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species")

  filtered <- ASV[, c("OTU_ID", taxonomy_cols, sample_cols)]

  mat <- filtered %>%
    select(-all_of(taxonomy_cols)) %>%
    column_to_rownames("OTU_ID") %>%
    t() %>%
    as.data.frame()

  list(matrix = mat, taxonomy = filtered[, c("OTU_ID", taxonomy_cols)])
}

MetaITS_root <- MetaITS_root %>%
  filter(sample_name %in% colnames(ASV_ITS))

MetaITS_rhizo <- MetaITS_rhizo %>%
```

```

filter(sample_name %in% colnames(ASV_ITS))

MetaITS_soil <- MetaITS_soil %>%
  filter(sample_name %in% colnames(ASV_ITS))

# --- STEP 3: Prepare ITS matrices ---
asvITS_root <- prepare_its_matrix(ASV_ITS, MetaITS_root)
asvITS_rhizo <- prepare_its_matrix(ASV_ITS, MetaITS_rhizo)
asvITS_soil <- prepare_its_matrix(ASV_ITS, MetaITS_soil)

# --- STEP 4: Run Indicator Species Analysis ---
run_isa_its <- function(matrix, meta) {
  group <- meta$Treatment[match(rownames(matrix), meta$sample_name)]
  multipatt(matrix, group, func = "r.g", control = how(nperm = 999))
}

isaITS_root <- run_isa_its(asvITS_root$matrix, MetaITS_root)
isaITS_rhizo <- run_isa_its(asvITS_rhizo$matrix, MetaITS_rhizo)
isaITS_soil <- run_isa_its(asvITS_soil$matrix, MetaITS_soil)

# --- STEP 5: Extract significant ITS species and clean ---
extract_its_sig <- function(isa_result, taxonomy, type) {
  as.data.frame(isa_result$sign) %>%
    rownames_to_column("OTU_ID") %>%
    filter(p.value <= 0.05) %>%
    left_join(taxonomy, by = "OTU_ID") %>%
    mutate(SampleType = type)
}

sigITS_root <- extract_its_sig(isaITS_root, asvITS_root$taxonomy, "Root")
sigITS_rhizo <- extract_its_sig(isaITS_rhizo, asvITS_rhizo$taxonomy, "Rhizo")
sigITS_soil <- extract_its_sig(isaITS_soil, asvITS_soil$taxonomy, "Soil")

sigITS_all <- bind_rows(sigITS_root, sigITS_rhizo, sigITS_soil)

# --- STEP 6: Clean labels and calculate relative abundance ---
sigITS_all <- sigITS_all %>%
  filter(
    !is.na(Species),
    !grepl("uncultured|metagenome|metagenomic|environmental", Species, ignore.case = TRUE),
    Species != ""
  ) %>%
  mutate(
    Treatment = case_when(
      s.control == 1 ~ "Control",
      s.E == 1 ~ "E",
      s.M == 1 ~ "M",
      s.ME == 1 ~ "ME"
    )
  )

calc_rel_abund_ITS <- function(matrix, taxonomy, metadata, type) {

```

```

rel_abund <- matrix / rowSums(matrix)

rel_long <- as.data.frame(t(rel_abund)) %>%
  rownames_to_column("OTU_ID") %>%
  pivot_longer(-OTU_ID, names_to = "sample_name", values_to = "rel_abundance") %>%
  left_join(taxonomy, by = "OTU_ID") %>%
  left_join(metadata, by = "sample_name") %>%
  group_by(Species, Treatment) %>%
  summarise(rel_abundance = mean(rel_abundance, na.rm = TRUE), .groups = "drop") %>%
  mutate(SampleType = type)

return(rel_long)
}

# --- STEP 7: Relative abundance for ITS ---
abITS_root <- calc_rel_abund_ITS(asvITS_root$matrix, asvITS_root$taxonomy, MetaITS_root, "Root")
abITS_rhizo <- calc_rel_abund_ITS(asvITS_rhizo$matrix, asvITS_rhizo$taxonomy, MetaITS_rhizo, "Rhizo")
abITS_soil <- calc_rel_abund_ITS(asvITS_soil$matrix, asvITS_soil$taxonomy, MetaITS_soil, "Soil")
abITS_all <- bind_rows(abITS_root, abITS_rhizo, abITS_soil)

# --- STEP 8: Merge & summarize ---
isaITS_merged <- sigITS_all %>%
  left_join(abITS_all, by = c("Species", "Treatment", "SampleType")) %>%
  group_by(Species, SampleType, Treatment) %>%
  summarise(MeanAbundance = mean(rel_abundance, na.rm = TRUE), .groups = "drop")

# --- STEP 9: Top 20 ITS indicator species
top20_ITS <- sigITS_all %>%
  group_by(Species) %>%
  summarise(TotalStat = sum(stat, na.rm = TRUE), .groups = "drop") %>%
  arrange(desc(TotalStat)) %>%
  slice_head(n = 20) %>%
  pull(Species)

isaITS_top20 <- isaITS_merged %>%
  filter(Species %in% top20_ITS)

# --- STEP 10: Pivot to wide format table
isaITS_summary_table <- isaITS_top20 %>%
  mutate(Treatment = factor(Treatment, levels = c("Control", "M", "E", "ME"))) %>%
  pivot_wider(
    names_from = c(SampleType, Treatment),
    values_from = MeanAbundance,
    values_fn = mean
  ) %>%
  arrange(Species)

isaITS_summary_table_percent <- isaITS_summary_table %>%
  mutate(across(where(is.numeric), ~ round(.x * 100, 2)))

# Export as CSV
write.csv(isaITS_summary_table_percent, "ISA_ITS_Top20_SummaryTable_Percent.csv", row.names = FALSE)

```

```

# --- Prepare data for plotting ---
isaITS_plot_data <- isaITS_merged %>%
  filter(Species %in% top20_ITS) %>%
  left_join(
    sigITS_all %>%
      select(Species, stat, p.value, SampleType, Treatment),
    by = c("Species", "SampleType", "Treatment")
  ) %>%
  mutate(Species_IndVal = paste0(Species, " (", round(stat, 2), ")"))

# --- Order species by IndVal descending ---
isaITS_plot_data$Species_IndVal <- factor(isaITS_plot_data$Species_IndVal,
  levels = isaITS_plot_data %>%
    group_by(Species_IndVal) %>%
    summarise(mean_IndVal = mean(stat, na.rm = TRUE)) %>%
    arrange(desc(mean_IndVal)) %>%
    pull(Species_IndVal)
)

# --- Ensure SampleType is in correct order for facets ---
isaITS_plot_data$SampleType <- factor(isaITS_plot_data$SampleType, levels = c("Root", "Rhizo", "Soil"))

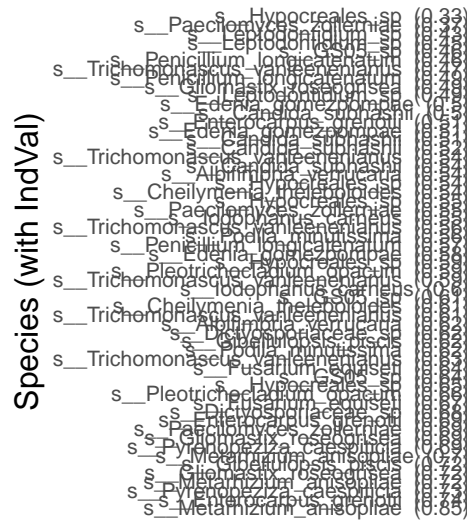
# --- Plot ---
ggplot(isaITS_plot_data, aes(x = Treatment, y = Species_IndVal)) +
  geom_point(aes(size = MeanAbundance * 100, fill = p.value), shape = 21, color = "black") +
  facet_wrap(~SampleType, scales = "free_x") +
  scale_size(range = c(1.5, 10), name = "Relative abundance (%)") +
  scale_fill_viridis_c(option = "plasma", direction = -1, name = "p-value", limits = c(0, 0.05)) +
  theme_minimal(base_size = 12) +
  labs(
    title = "Indicator Species Analysis (ITS): Top 20 Significant Species",
    x = "Treatment Group (Control, M, E, ME)",
    y = "Species (with IndVal)"
  ) +
  theme(
    axis.text.y = element_text(size = 8),
    legend.position = "bottom"
  )


```

```

## Warning: Removed 12 rows containing missing values or values outside the scale range
## ('geom_point()').

```

[illegible]

p-value  0.0000