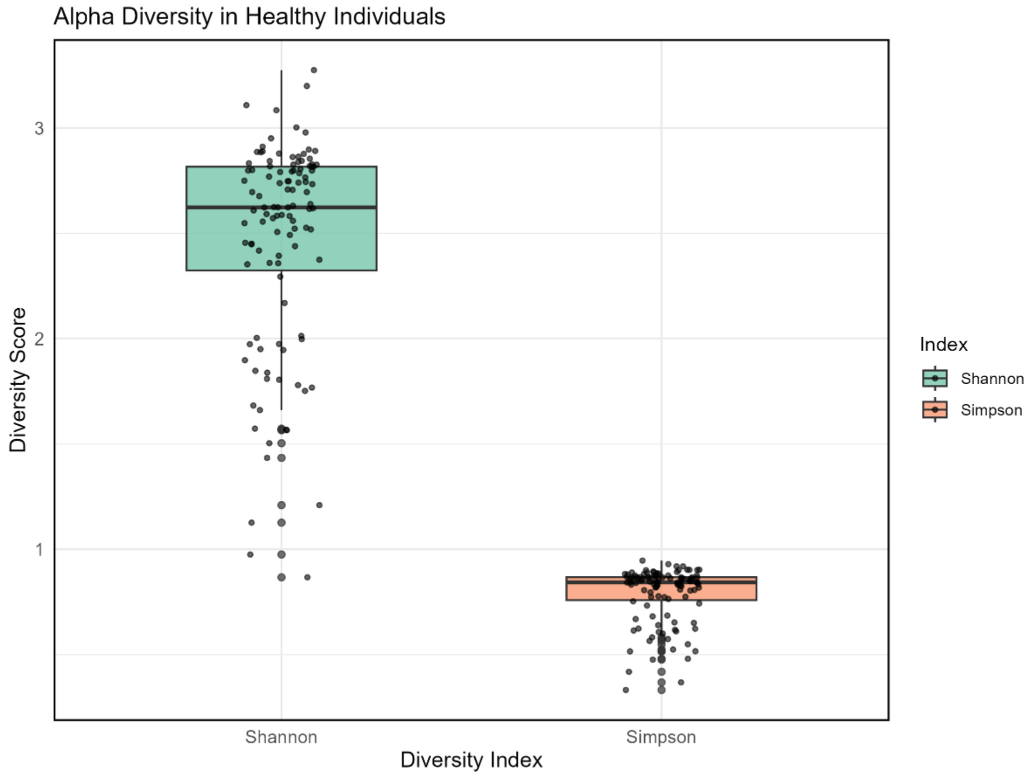
Codes for My analysis Tutorial

1. **Code for alpha-diversity of HH**

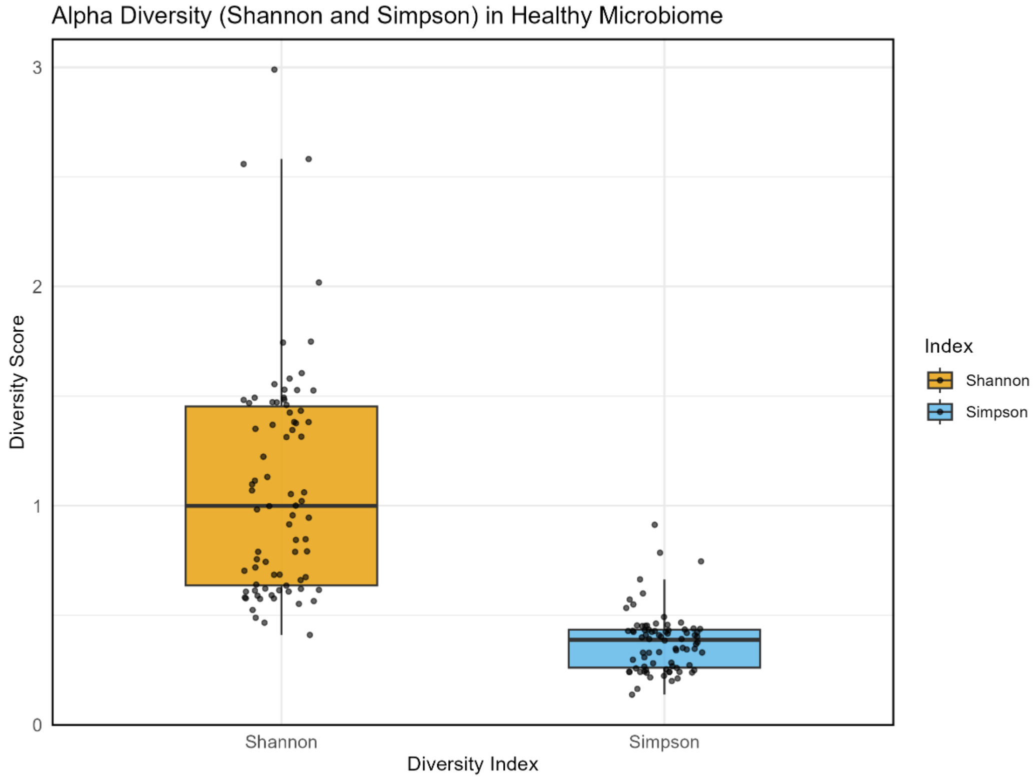
```
#Starting From the Beggenning
hh_species<-
read.csv("https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/HH_combined_bracken_species_fraction.csv", row.names = 1, check.names = FALSE)
view(hh_species)
#Transpose
hh_species_t<-t(hh_species)
view(hh_species_t)
#Now:
#rownames(hh_species_t) = sample IDs
#colnames(hh_species_t) = species names ✅
#Convert to Numeric Safely without losing names
#Creat a new data frame, retaining column and row names
hh_species_t_numeric<-as.data.frame(hh_species_t)
#use lapply to convert each column to numeric while preserving names
hh_species_t_numeric<-hh_species_t_numeric%>%
  mutate(across(everything(), ~as.numeric(.)))
#check that species names are still in column names
head(colnames(hh_species_t_numeric))
#Check Data Structure
str(hh_species_t_numeric)
#Lets compute Diversity indices
library(vegan)
shannon_index_species<-diversity(hh_species_t_numeric,index = "shannon")
simpson_index_species<-diversity(hh_species_t_numeric, index = "simpson")

#Prepare Data for Plotting
diversity_df <- data.frame(
  SampleID = rownames(hh_species_t_numeric),
  Shannon = shannon_index_species,
  Simpson = simpson_index_species
)

#Convert this long Plot format for easier plotting:
library(tidyr)

diversity_long_species <- pivot_longer(
  diversity_df,
  cols = c("Shannon", "Simpson"),
  names_to = "Index",
  values_to = "Value"
)
view(diversity_long_species)

#Box Plot 2
library(ggplot2)

p <- ggplot(diversity_long, aes(x = Index, y = Value, fill = Index)) +
  geom_boxplot(width = 0.5, alpha = 0.7) +
  geom_jitter(width = 0.1, alpha = 0.6, color = "black", size = 1) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12)
  ) +
  labs(
    title = "Alpha Diversity in Healthy Individuals",
    x = "Diversity Index",
    y = "Diversity Score"
  ) +
  scale_fill_manual(values = c("Shannon" = "#66c2a5", "Simpson" = "#fc8d62"))
# Show the plot
print(p)
```



2. Code #Alpha Diversity Calculations Computation for pC

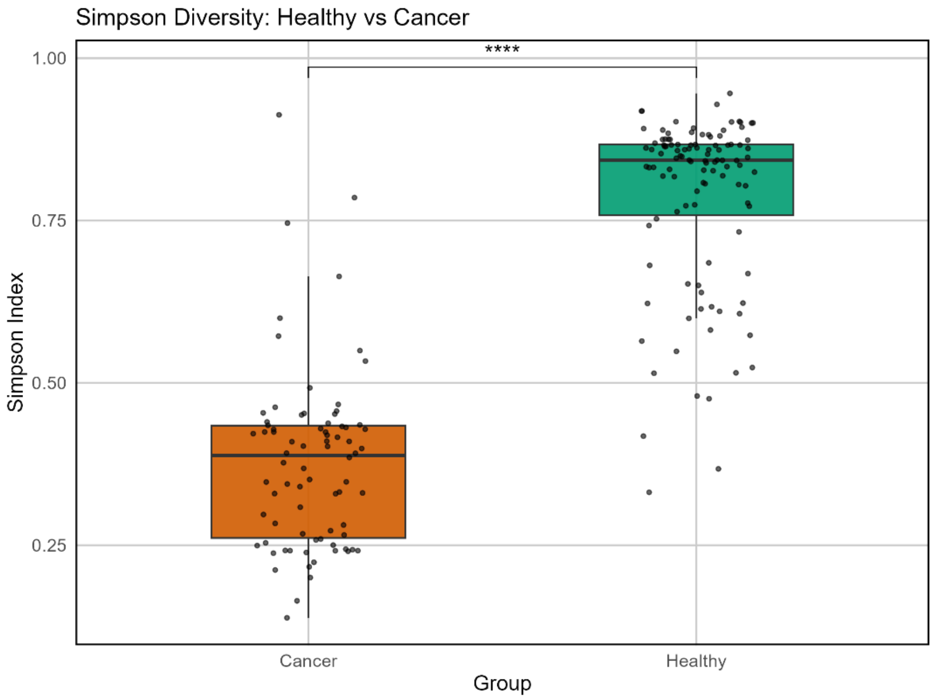
```
pC_species<-
read.csv("https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/PC_combined_bracken_species_fraction.csv", row.names = 1, check.names = FALSE)
view(pC_species)
#Transpose
pC_species_t<-t(pC_species)
view(pC_species_t)
#Rownames(pC_species_t) =sample ID
#Colnames(pC_species_t) = species name
#Convert to numeric safely without losing names
#Create a new data frame, retaining column and row names
pC_species_t_numeric<-as.data.frame(pC_species_t)
view(pC_species_t_numeric)
#Use lapply to convert each column to numeric while preserving names
pC_species_t_numeric<-pC_species_t_numeric%>%
  mutate(across(everything(),~as.numeric(.)))
#Check that species names are still in column names
head(colnames(pC_species_t_numeric))
#Check Data Structure
str(pC_species_t_numeric)
#Lets compute DIversity Indices
library(vegan)
pC_shannon_index_sp<-diversity(pC_species_t_numeric, index = "shannon")
pC_simpson_index_sp<-diversity(pC_species_t_numeric, index = "simpson")
#Prepare Data for Plotting
pC_diversity_df<-data.frame(SampleID = rownames(pC_species_t_numeric),Shannon =
pC_shannon_index_sp, Simpson = pC_simpson_index_sp)
view(pC_diversity_df)
#Convert this long plot format for easier plotting:
library(tidyr)
pC_diversity_long_sp<-pivot_longer(pC_diversity_df, cols = c("Shannon", "Simpson"), names_to = "Index",
values_to ="Value")
view(pC_diversity_long_sp)
#Box Plot
library(ggplot2)
pC_sp <- ggplot(pC_diversity_long_sp, aes(x = Index, y = Value, fill = Index)) +
  geom_boxplot(width = 0.5, alpha = 0.8, outlier.shape = NA) +
  geom_jitter(width = 0.1, alpha = 0.6, color = "black", size = 1) +
  theme_minimal() +
  theme(
    axis.text = element_text(size = 10),
    panel.border = element_rect(color = "black", fill = NA, size = 1)
  ) +
  labs(
    title = "Alpha Diversity (Shannon and Simpson) in Healthy Microbiome",
    x = "Diversity Index",
    y = "Diversity Score"
  ) +
  scale_fill_manual(values = c("Shannon" = "#E69F00", "Simpson" = "#56B4E9"))  # color-blind friendly
print(pC_sp)
```

```
ggsave("alpha_diversity_boxplot.png", plot = pC_sp, width = 8, height = 6, dpi = 300)
```



3. Code: #Comparative Diversity Plots for HH & pC

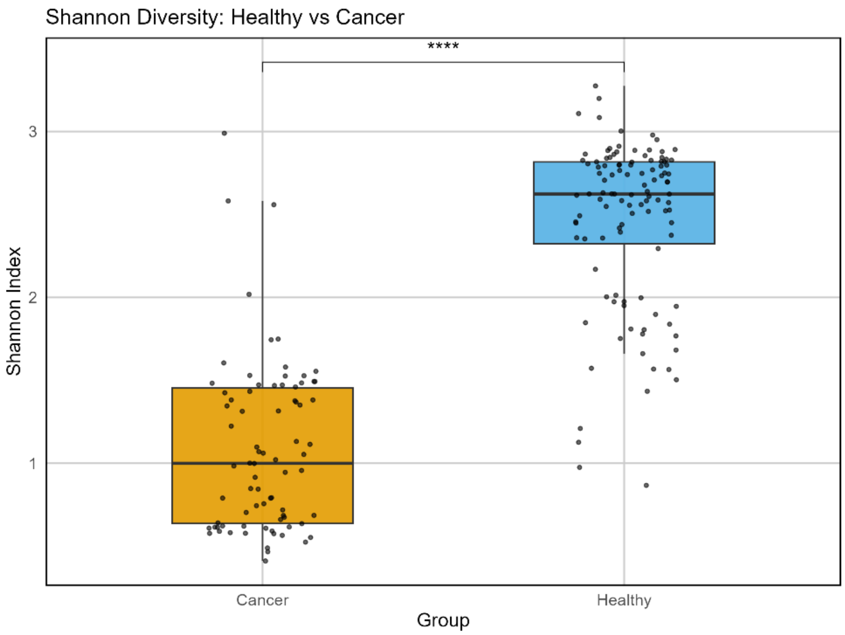
```
install.packages(c("readxl", "vegan", "ggplot2", "dplyr", "ggpubr"))
library(readxl)
library(vegan)
library(ggplot2)
library(dplyr)
library(ggpubr)
view(diversity_df)
view(pC_diversity_df)
#Group Data
pC_diversity_df$Group<-"Cancer"
diversity_df$Group<-"Healthy"
#Combine Both Datasets
combined_div<-rbind(diversity_df, pC_diversity_df)
view(combined_div)
#Plot: Simpson BOXplot with Statistical Test
library(ggplot2)
library(ggpubr)

p_simpson <- ggplot(combined_div, aes(x = Group, y = Simpson, fill = Group)) +
  geom_boxplot(width = 0.5, alpha = 0.9, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.6, size = 1, color = "black") +

  # Statistical test with bar and asterisks
  stat_compare_means(method = "wilcox.test", label = "p.signif",
                     comparisons = list(c("Healthy", "Cancer")),
                     tip.length = 0.02, size = 5) +

  # Apply a clean theme and add gridlines + border
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    axis.text = element_text(size = 11),
    axis.title = element_text(size = 13),
    legend.position = "none"
  ) +

  labs(
    title = "Simpson Diversity: Healthy vs Cancer",
    x = "Group",
    y = "Simpson Index"
  ) +

  # Color-blind friendly & elegant colors (Okabe-Ito palette)
  scale_fill_manual(values = c("Healthy" = "#009E73", "Cancer" = "#D55E00"))
print(p_simpson)
ggsave("combined_Simpson_Diversity_Boxplot.png", plot = p_simpson, width = 8, height = 6, dpi = 300)

# Wilcoxon rank-sum test (non-parametric)
wilcox.test(Simpson ~ Group, data = combined_div)
```



```
	Wilcoxon rank sum test wit
h continuity correction

data:  Simpson by Group
W = 327, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

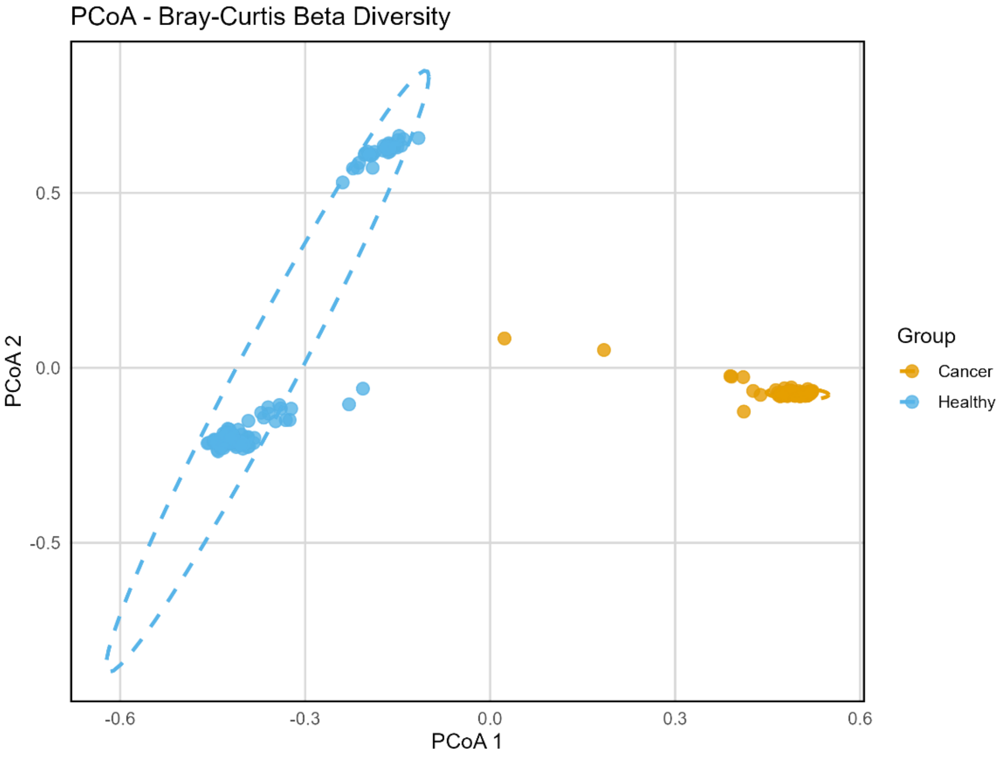
4. Combined for Shannon Index:

```
p_shannon <- ggplot(combined_div, aes(x = Group, y = Shannon, fill = Group)) +
  geom_boxplot(width = 0.5, alpha = 0.9, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.6, size = 1, color = "black") +
  # Statistical test with bar and asterisks
  stat_compare_means(method = "wilcox.test", label = "p.signif",
                     comparisons = list(c("Healthy", "Cancer")),
                     tip.length = 0.02, size = 5) +

  # Apply a clean theme and add gridlines + border
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    axis.text = element_text(size = 11),
    axis.title = element_text(size = 13),
    legend.position = "none"
  ) +

  labs(
    title = "Shannon Diversity: Healthy vs Cancer",
    x = "Group",
    y = "Shannon Index"
  ) +

  # Color-blind friendly & elegant colors (Okabe-Ito palette)
  scale_fill_manual(values = c("Healthy" = "#56B4E9", "Cancer" = "#E69F00"))
print(p_shannon)
ggsave("combined_Simpson_Diversity_Boxplot.png", plot = p_shannon, width = 8, height = 6, dpi = 300)

# Wilcoxon rank-sum test (non-parametric)
wilcox.test(Shannon ~ Group, data = combined_div)
```



```
Wilcoxon                                                                  rank sum test with conti
nuity correction

data:  Shannon by Group
W = 396, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

BetaDIversity: Step 3: Plot PCoA with group ellipses (optional)



Code:

```
#Beta-Diversity
library(readxl)
library(vegan)
library(ggplot2)
library(dplyr)
hh_df_sp<-as.data.frame(hh_species_t)
pC_df_sp<-as.data.frame(pC_species_t)
hh_df_sp$Group<-"Healthy"
view(hh_df_sp)
view(pC_diversity_df)
pC_df_sp$Group<-"Cancer"
#Combine and Clean Data
#First, since the species are different in the different datset, (Force matching is used)
#1: Get the union of all species names (columns)
all_species_beta<-union(colnames(hh_df_sp), colnames(pC_df_sp))
#Add missing Columns (species) with zeros to cancer data
missing_in_pC <- setdiff(all_species_beta, colnames(pC_df_sp))
pC_df_sp[missing_in_pC] <- 0
#Re-order columns to match correctly
hh_df_sp <- hh_df_sp[, all_species_beta]
pC_df_sp <- pC_df_sp[, all_species_beta]
#Reorder before combining
# Fill missing species with zeros (not yet reordered)
missing_in_hh <- setdiff(all_species_beta, colnames(hh_df_sp))
missing_in_pc <- setdiff(all_species_beta, colnames(pC_df_sp))

hh_df_sp[missing_in_hh] <- 0
pC_df_sp[missing_in_pc] <- 0
#Order to Match
hh_df_sp <- hh_df_sp[, all_species_beta]
pC_df_sp <- pC_df_sp[, all_species_beta]
view(hh_df_sp)
#Group
pC_df_sp$Group<-"Cancer"
hh_df_sp$Group<-"Healthy"
#Combine
combined_betadiv_df <- rbind(hh_df_sp, pC_df_sp)
view(combined_betadiv_df)
hh_df_sp$Group <- "Healthy"
pC_df_sp$Group <- "Cancer"
view(hh_df_sp)
#After combining
group_labels <- combined_betadiv_df$Group
combined_betadiv_df$Group <- NULL  # remove Group column before calculating distances
view(group_labels)
#Calculate Bray-Cutis distance
library(vegan)

# Make sure group labels are stored
group_labels <- c(rep("Healthy", nrow(hh_df_sp)), rep("Cancer", nrow(pC_df_sp)))

# Combine numeric species tables (already fixed earlier)
combined_betadiv_df <- rbind(hh_df_sp, pC_df_sp)

# Calculate Bray-Curtis distance matrix
```

```
bray_dist <- vegdist(combined_betadiv_df, method = "bray")

#Check data frame:
str(combined_betadiv_df)

#STore Groups
group_labels <- combined_betadiv_df$Group

#Remove group columns before computing distances
# Keep only numeric species data
combined_numeric_df <- combined_betadiv_df[, sapply(combined_betadiv_df, is.numeric)]
#Compute Bryacutis
library(vegan)
bray_dist <- vegdist(combined_numeric_df, method = "bray")
# Run PCoA (Principal Coordinates Analysis)
pcoa_result <- cmdscale(bray_dist, eig = TRUE, k = 2)

# Create a data frame for plotting
pcoa_df <- data.frame(
  SampleID = rownames(combined_betadiv_df),
  Dim1 = pcoa_result$points[, 1],
  Dim2 = pcoa_result$points[, 2],
  Group = group_labels
)
library(ggplot2)

# Create the plot and assign it to an object
pcoa_plot <- ggplot(pcoa_df, aes(x = Dim1, y = Dim2, color = Group)) +
  geom_point(size = 3, alpha = 0.8) +
  stat_ellipse(level = 0.95, linetype = "dashed", size = 1) +  # 95% CI ellipse
  theme_minimal(base_size = 12) +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_line(color = "grey85"),
    panel.grid.minor = element_blank(),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    legend.position = "right"
  ) +
  labs(
    title = "PCoA - Bray-Curtis Beta Diversity",
    x = "PCoA 1",
    y = "PCoA 2"
  ) +
  scale_color_manual(values = c("Healthy" = "#56B4E9", "Cancer" = "#E69F00"))

# Print the plot
print(pcoa_plot)
ggsave("PCoA_BrayCurtis_BetaDiversity.png", plot = pcoa_plot, width = 8, height = 6, dpi = 300)
```

Code, Beta Diversity: **Option 2: PERMANOVA (adonis2)**

Goal: Statistically test whether microbial composition differs between groups.

```
Permutation test for adonis under reduced model
Permutation: free
Number of permutations: 999

adonis2(formula = bray_dist ~ Group, data = metadata_df, permutations = 999)
          Df SumOfSqs      R2      F Pr(>F)    
Model      1   31.535 0.52997 210.84  0.001 ***
Residual 187   27.969 0.47003                  
Total    188   59.504 1.00000                  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>
```

- **R^2** = proportion of variation explained by group
- **p-value** (`Pr(>F)`) = significance (if < 0.05, groups are significantly different

Tool	Use in literature	Purpose
PCoA + Bray-Curtis	✓ Common for visualization	Show group clustering/distance
PERMANOVA (adonis2)	✓✓ Common for testing	Quantify if group differences are significant

PART B: **comparative taxonomic analysis**,

Research Questions:

Goal	Research Question
1□	Which bacteria are **significantly more abundant** in Healthy vs Cancer?
2□	Which bacteria **overlap** between groups?
3□	Which bacteria are **strongly associated** with Healthy or Cancer groups (as potential indicators)?

Analysis Type	What it Answers	Method/Tool	Plot Suggestion
1. **Differential Abundance**	What taxa are more abundant in one group?	Wilcoxon test, `DESeq2`, `ALDEx2`	Volcano plot, boxplot
2. **Venn Diagram**	What taxa overlap vs unique?	Base R + `VennDiagram`	Venn diagram

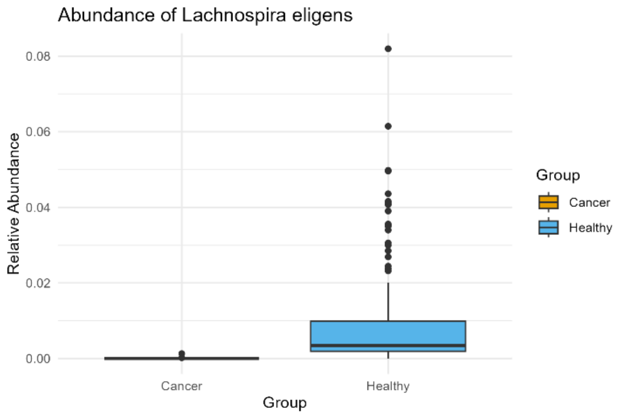
Analysis Type	What it Answers	Method/Tool	Plot Suggestion
3. **Indicator Species Analysis / Biomarker Discovery**	Which taxa are predictive of group?	`indicspecies`, `LEfSe`, `Random Forest`	Dot plot or importance plot
4. **Bar Plot/Heatmap**	Visualize abundance patterns	`ggplot2`, `pheatmap`	Stacked bar, heatmap

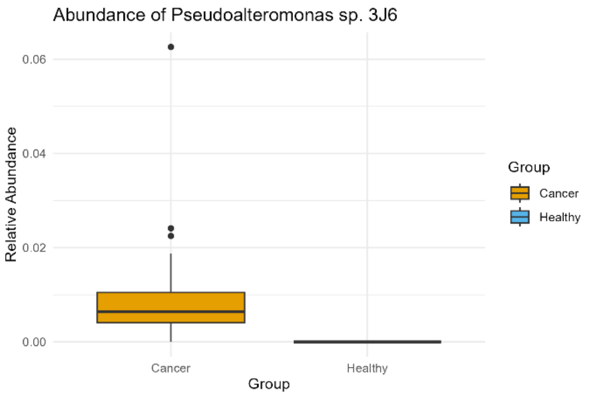
1. Relative abundance for the top 5 microbes in Healthy VS disease patients

```
#PART B Taxanomic COmparison
#Q: Which Bacteria are more Abundant in Healthy VS Cancer?
#Option A: Wilcoxon test (for small datasets)
#Combine data
install.packages(c("readxl", "dplyr", "ggplot2", "VennDiagram", "indicspecies"))
install.packages(c("VennDiagram", "indicspecies"))
library(readxl)
library(dplyr)
library(ggplot2)
library(VennDiagram)
library(indicspecies)

#STEP 2: Read Excel Data (Relative Abundance Tables)
library(readr)

# Re-import correctly
# Load necessary library
library(readr)

# Load required library
library(ggplot2)

cat("=== Step 1: Reading and cleaning Healthy data ===\n")
h_raw <- read.csv(

"https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/HH_combined_bracken_species_fraction.csv",
  row.names = 1, check.names = FALSE
)
h_raw_clean <- h_raw[!rownames(h_raw) %in% c("Homo sapiens"), ]
h_t_clean <- as.data.frame(t(h_raw_clean))
h_t_clean[] <- lapply(h_t_clean, function(x) as.numeric(as.character(x)))
h_t_clean$Group <- "Healthy"
cat("✅ Healthy dataset cleaned and transposed.\n")

cat("=== Step 2: Reading and cleaning Cancer data ===\n")
p_raw <- read.csv(

"https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/PC_combined_bracken_species_fraction.csv",
  row.names = 1, check.names = FALSE
)
p_raw_clean <- p_raw[!rownames(p_raw) %in% c("Homo sapiens"), ]
p_t_clean <- as.data.frame(t(p_raw_clean))
p_t_clean[] <- lapply(p_t_clean, function(x) as.numeric(as.character(x)))
p_t_clean$Group <- "Cancer"
cat("✅ Cancer dataset cleaned and transposed.\n")

cat("=== Step 3: Harmonizing and merging datasets ===\n")
```

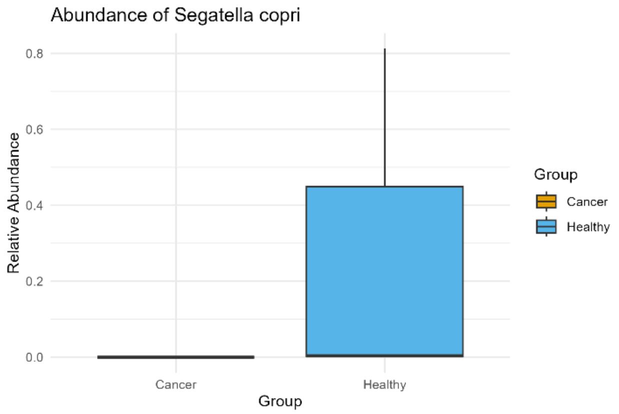
```
all_species <- union(colnames(h_t_clean), colnames(p_t_clean))
missing_in_hh <- setdiff(all_species, colnames(h_t_clean))
missing_in_pc <- setdiff(all_species, colnames(p_t_clean))
h_t_clean[missing_in_hh] <- 0
p_t_clean[missing_in_pc] <- 0
h_t_clean <- h_t_clean[, all_species]
p_t_clean <- p_t_clean[, all_species]
combined_data <- rbind(h_t_clean, p_t_clean)
cat("✅ Combined dataset created. Structure:\n")
print(str(combined_data))

cat("=== Step 4: Preparing for Wilcoxon testing ===\n")
group_labels <- combined_data$Group
abundance_data <- combined_data[, !colnames(combined_data) %in% "Group"]
cat("✅ Confirm Homo sapiens excluded: ", "Homo sapiens" %in% colnames(abundance_data), "\n")

cat("=== Step 5: Running Wilcoxon tests ===\n")
pvals <- apply(abundance_data, 2, function(x) {
  wilcox.test(x ~ group_labels)$p.value
})
padj <- p.adjust(pvals, method = "fdr")
results_df <- data.frame(
  Species = names(pvals),
  p_value = pvals,
  p_adj = padj
)
cat("✅ Wilcoxon test completed. Top species:\n")
print(head(results_df[order(results_df$p_adj), ]))

cat("=== Step 6: Plotting top 5 species ===\n")

top_species <- results_df$Species[order(results_df$p_adj)][1:5]
```

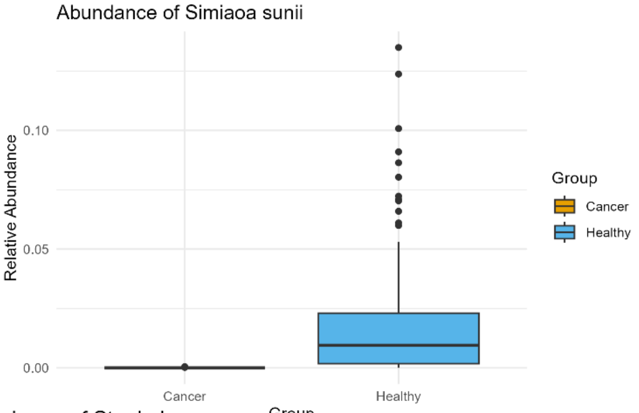
Generate Graphs:

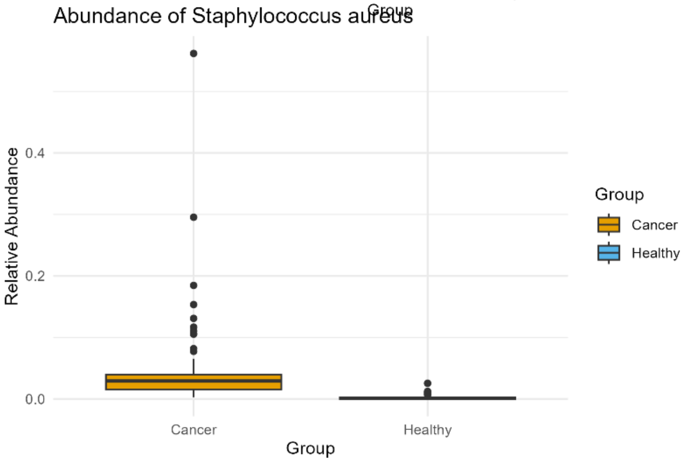




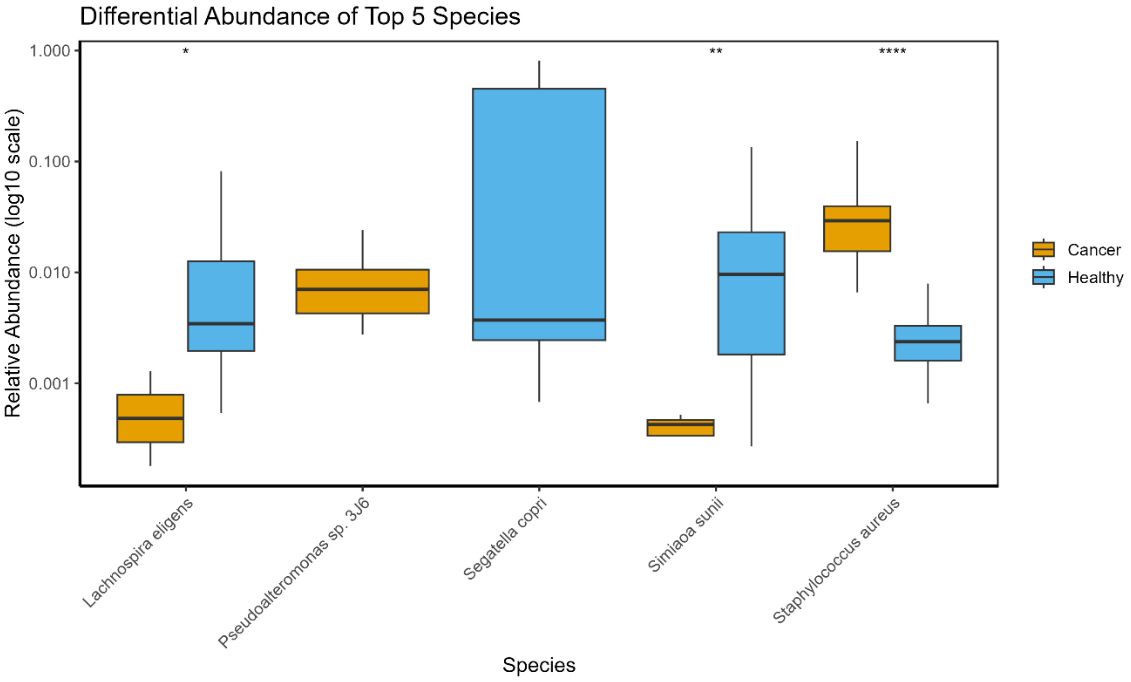
```
# Load ggplot2 if not already loaded
library(ggplot2)

# Loop through top species and save each plot
for (sp in top_species) {
  p <- ggplot(combined_data, aes(x = Group, y = .data[[sp]], fill = Group)) +
    geom_boxplot() +
    theme_minimal() +
    labs(title = paste("Abundance of", sp), y = "Relative Abundance") +
    scale_fill_manual(values = c("Healthy" = "#56B4E9", "Cancer" = "#E69F00"))

  # Generate a safe filename
  fname <- paste0("boxplot_", gsub("[^a-zA-Z0-9]", "_", sp), ".png")

  # Save the plot
  ggsave(filename = fname, plot = p, width = 6, height = 4, dpi = 300)

  # Optional: print confirmation
  cat("✅ Saved:", fname, "\n")
}
```





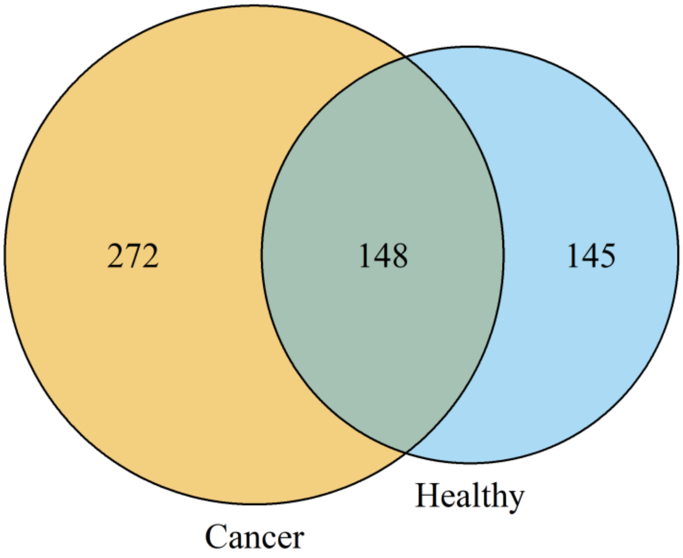


CODE for GROUPED Graph



```
# Load necessary libraries
library(ggplot2)
library(tidyr)
library(dplyr)
library(ggpubr)

# Step 1: Select top 5 species
top_species <- results_df$Species[order(results_df$p_adj)][1:5]

# Step 2: Subset and reshape data
abundance_subset <- combined_data[, c(top_species, "Group")]

abundance_long <- pivot_longer(
  abundance_subset,
  cols = all_of(top_species),
  names_to = "Species",
  values_to = "Abundance"
)

# Step 3: Plot with enclosed axes and stat bar
p <- ggplot(abundance_long, aes(x = Species, y = Abundance, fill = Group)) +
  geom_boxplot(position = position_dodge(0.8), outlier.shape = NA) +
  stat_compare_means(
    aes(group = Group),
    method = "wilcox.test",
    label = "p.signif",
    label.y.npc = "top",
    bracket.size = 0.6,
    tip.length = 0.02,
    size = 4,
    hide.ns = TRUE
  ) +
  scale_y_continuous(trans = "log10") +
  scale_fill_manual(values = c("Healthy" = "#56B4E9", "Cancer" = "#E69F00")) +
  labs(
    title = "Differential Abundance of Top 5 Species",
    x = "Species",
    y = "Relative Abundance (log10 scale)"
  ) +
  theme_classic(base_size = 13) +  # Use classic theme for boxed plot
  theme(
    axis.line = element_line(color = "black", size = 0.8),
    panel.border = element_rect(fill = NA, color = "black", size = 1),  # Full box around plot
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right",       # Legend on the right
    legend.title = element_blank()
  )

# Step 4: Print to viewer
print(p)

# Step 5: Save to PNG
ggsave("top5_species_boxplot.png", plot = p, width = 10, height = 6, dpi = 300)
cat("✅ Plot saved as 'top5_species_boxplot.png'\n")
```

Venn Diagram:



Codes:

```
library(VennDiagram)

# Step 1: Identify numeric columns only
hh_numeric_cols <- sapply(h_t_clean, is.numeric)
pc_numeric_cols <- sapply(p_t_clean, is.numeric)
```

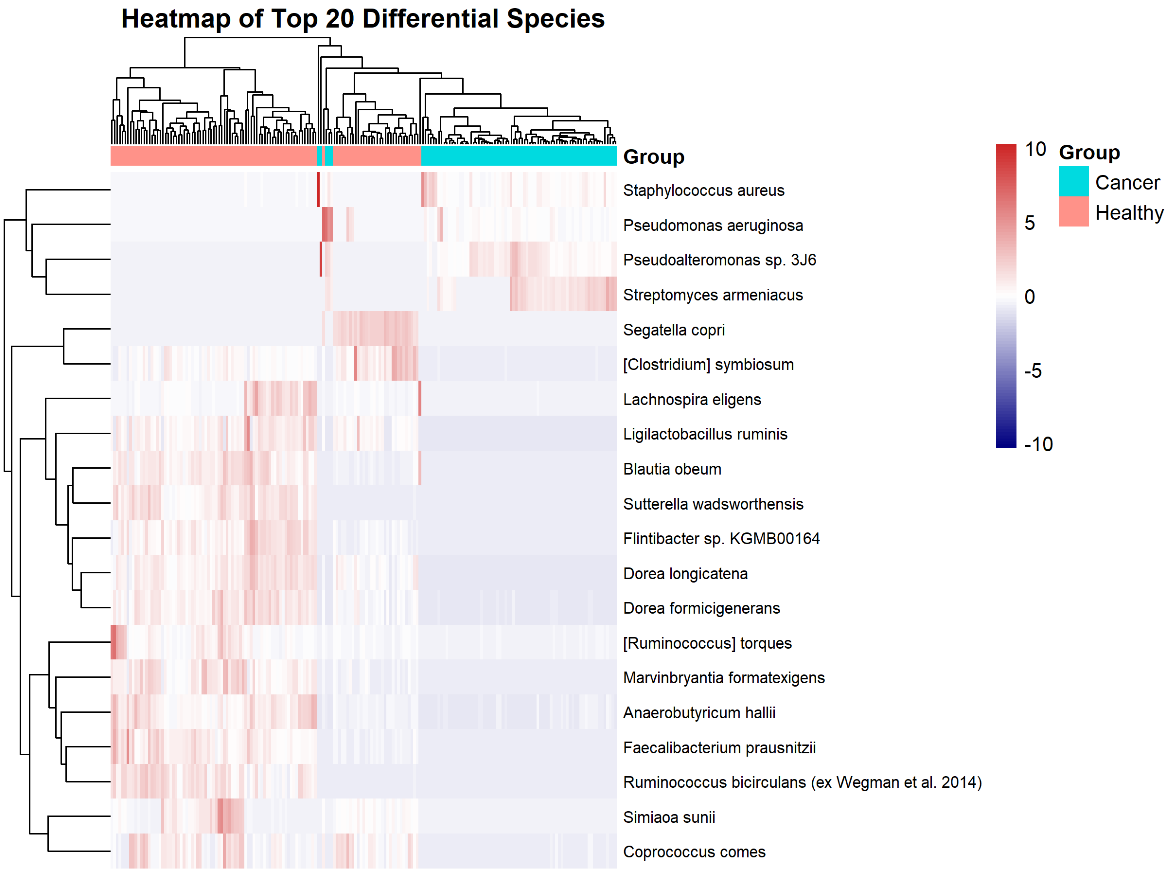
```
# Step 2: Get species (columns) with non-zero abundance
hh_species <- colnames(h_t_clean)[hh_numeric_cols][colSums(h_t_clean[, hh_numeric_cols]) > 0]
pc_species <- colnames(p_t_clean)[pc_numeric_cols][colSums(p_t_clean[, pc_numeric_cols]) > 0]

# Step 3: Generate and save Venn diagram
venn.plot <- venn.diagram(
  x = list(Healthy = hh_species, Cancer = pc_species),
  category.names = c("Healthy", "Cancer"),
  filename = "venn_species_overlap.png",  # Saves to current working directory
  output = TRUE,
  imagetype = "png",
  height = 2000,
  width = 2000,
  resolution = 300,
  col = "black",
  fill = c("#56B4E9", "#E69F00"),
  alpha = 0.5,
  cex = 2,
  cat.cex = 2,
  cat.pos = 0
)
```

HeatMap:

```
library(pheatmap)

# Choose top 20 species based on lowest adjusted p-values
top_heatmap_species <- results_df$Species[order(results_df$p_adj)][1:20]

# Prepare abundance matrix for heatmap (samples as rows, species as columns)
```

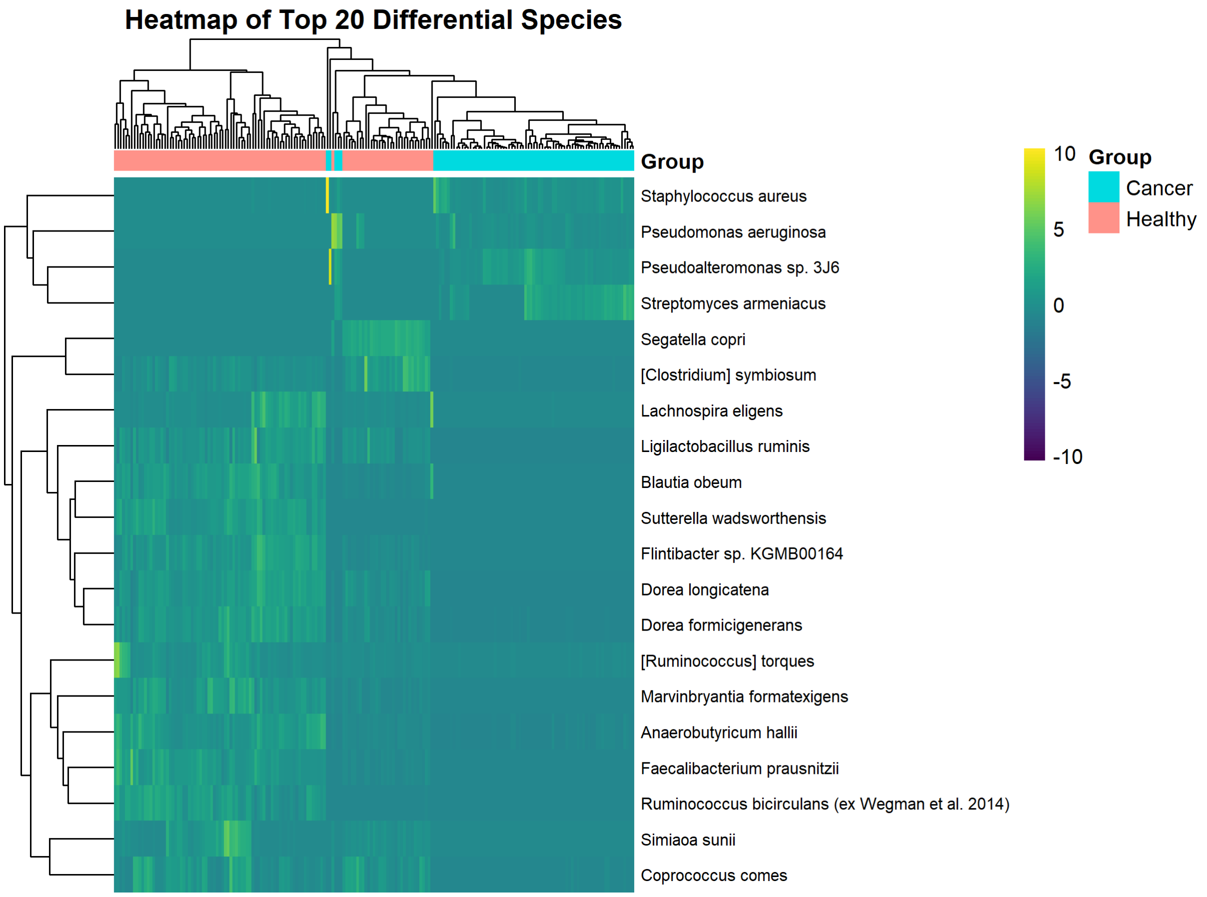
```
heatmap_data <- combined_data[, top_heatmap_species]
rownames(heatmap_data) <- paste0(combined_data$Group, "_", seq_len(nrow(combined_data)))

# Group annotation (used for coloring rows by Healthy or Cancer)
annotation <- data.frame(Group = combined_data$Group)
rownames(annotation) <- rownames(heatmap_data)

# Plot heatmap with species names shown (they are columns)
pheatmap(
  t(heatmap_data),                # Transpose so species are on y-axis
  annotation_col = annotation,   # Now columns are samples
  scale = "row",                 # Normalize species abundance per row
  show_rownames = TRUE,          # Show species names
  show_colnames = FALSE,         # Hide sample names if too many
  fontsize_row = 8,              # Adjust if species names are too long
  clustering_distance_rows = "euclidean",
  clustering_distance_cols = "euclidean",
  color = colorRampPalette(c("navy", "white", "firebrick3"))(100),
  main = "Heatmap of Top 20 Differential Species"
)
```

Code: to plot the heat map graph:



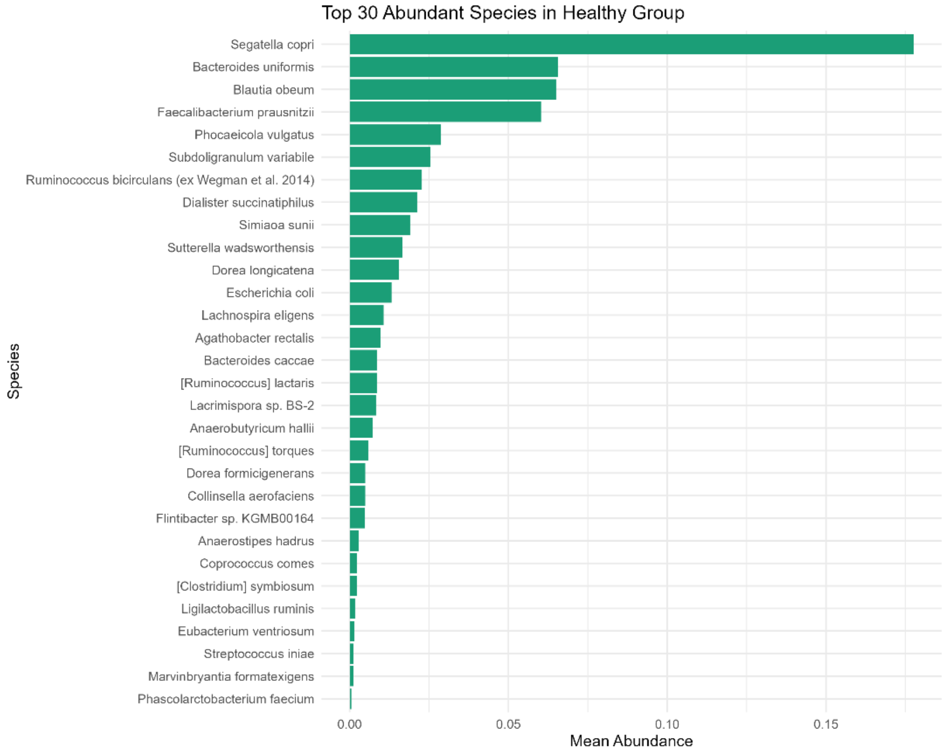
c

Code to plot color blind heat map:

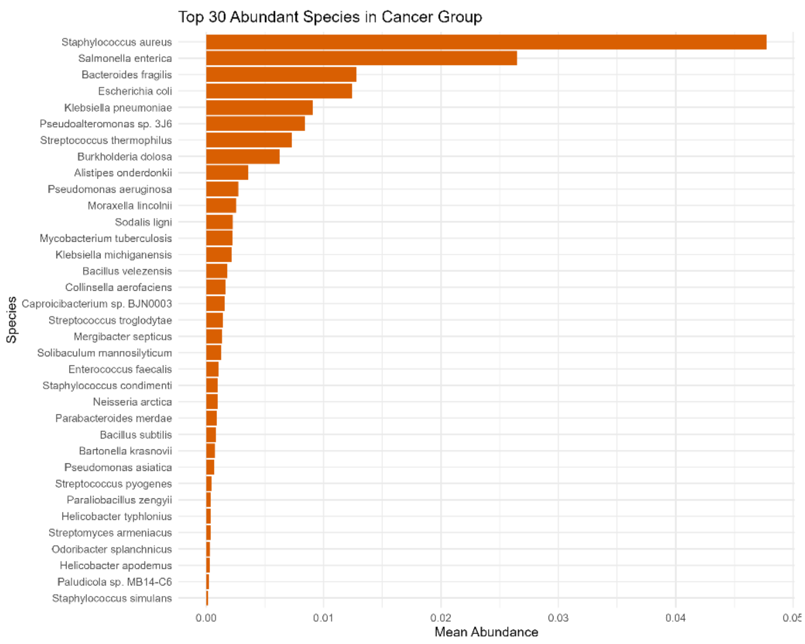


```
library(pheatmap)
library(viridis)

pheatmap(
  t(heatmap_data),
  annotation_col = annotation,
  scale = "row",
  show_rownames = TRUE,
  show_colnames = FALSE,
  fontsize_row = 8,
  clustering_distance_rows = "euclidean",
  clustering_distance_cols = "euclidean",
  color = viridis(100, option = "D"),  # Use viridis palette
  main = "Heatmap of Top 20 Differential Species",
  filename = "top20_species_heatmap_colorblind_friendly.png",
  width = 8, height = 6
)
```

Index species for both groups:



```
                   #Top 30 Gut microbiome

# Load libraries
library(dplyr)
library(ggplot2)

# Step 1: Prepare the abundance matrix and group info
group_factor <- combined_data$Group

# ✅ Remove non-numeric columns for abundance matrix
abundance_matrix <- combined_data[, sapply(combined_data, is.numeric)]

# Step 2: Run indicator species analysis
library(indicspecies)
library(vegan)
abundance_matrix <- combined_data[, -ncol(combined_data)]  # assuming last column is Group
ind_result <- multipatt(abundance_matrix, group_factor, control = how(nperm = 999))
group_factor <- combined_data$Group

# Step 2: Run indicator species analysis
library(indicspecies)
library(vegan)
ind_result <- multipatt(abundance_matrix, group_factor, control = how(nperm = 999))

# Step 3: Extract significant species
ind_df <- as.data.frame(ind_result$sign)
ind_df$Species <- rownames(ind_df)

# Step 4: Identify top 30 species per group based on indicator statistic
top_healthy <- ind_df %>%
  filter(s.Healthy == 1) %>%
  slice_max(stat, n = 30)

top_cancer <- ind_df %>%
  filter(s.Cancer == 1) %>%
  slice_max(stat, n = 30)

# Step 5: Get species names
top_healthy_species <- top_healthy$Species
top_cancer_species <- top_cancer$Species

# Step 6: Subset original data by group
healthy_data <- combined_data %>% filter(Group == "Healthy")
cancer_data  <- combined_data %>% filter(Group == "Cancer")

# Step 7: Extract abundance values for top species
healthy_abundances <- healthy_data[, top_healthy_species]
cancer_abundances  <- cancer_data[, top_cancer_species]

# Step 8: Compute mean abundance per species
healthy_means <- colMeans(healthy_abundances, na.rm = TRUE)
cancer_means  <- colMeans(cancer_abundances, na.rm = TRUE)

df_healthy <- data.frame(Species = names(healthy_means), Abundance = healthy_means, Group = "Healthy")
df_cancer  <- data.frame(Species = names(cancer_means), Abundance = cancer_means, Group = "Cancer")


# Step 9: Plot bar graphs
p_healthy <- ggplot(df_healthy, aes(x = reorder(Species, Abundance), y = Abundance)) +
  geom_bar(stat = "identity", fill = "#1b9e77") +
  coord_flip() +
  labs(title = "Top 30 Abundant Species in Healthy Group", x = "Species", y = "Mean Abundance") +
  theme_minimal(base_size = 12)
```

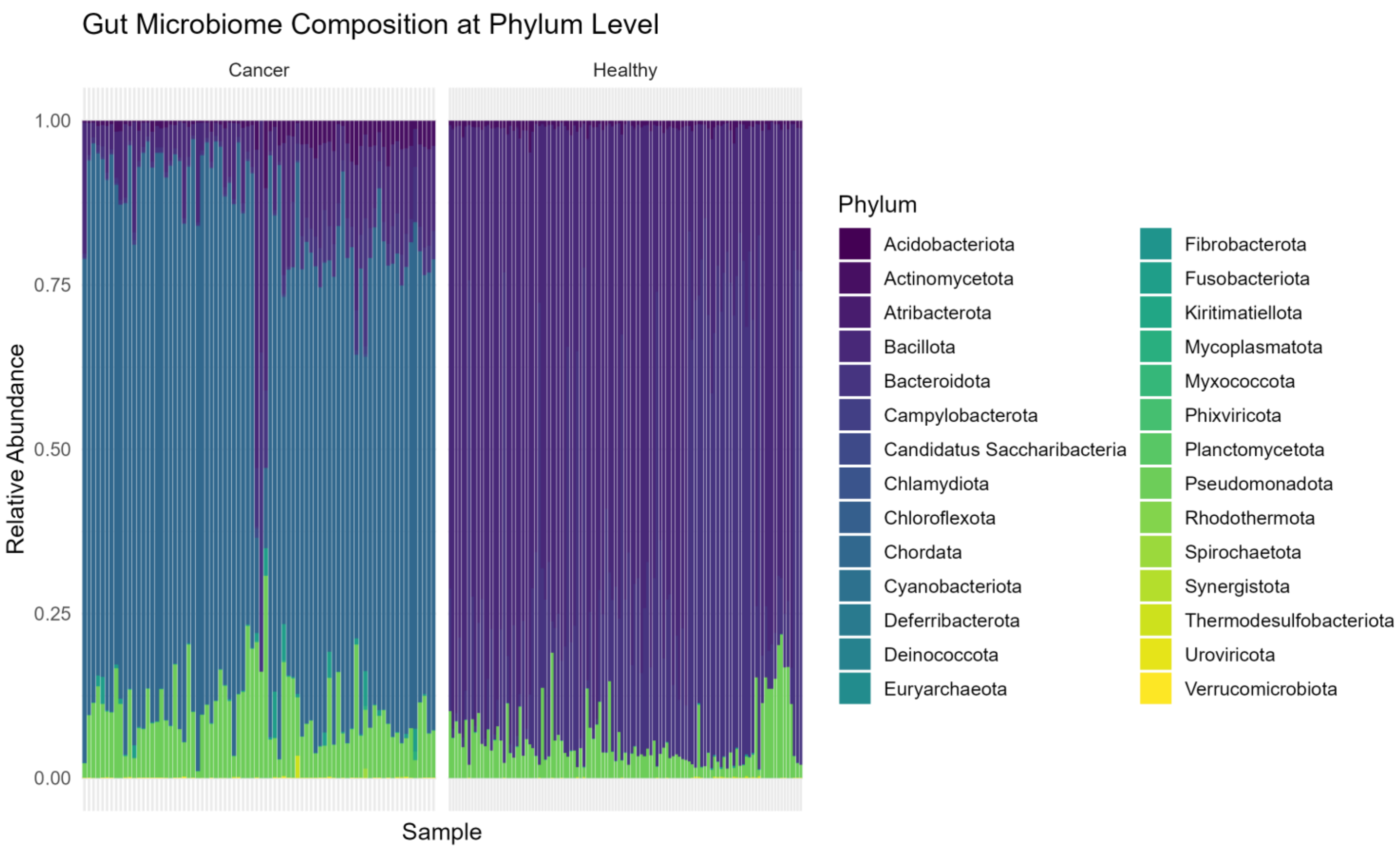


```
p_cancer <- ggplot(df_cancer, aes(x = reorder(Species, Abundance), y = Abundance)) +
  geom_bar(stat = "identity", fill = "#d95f02") +
  coord_flip() +
  labs(title = "Top 30 Abundant Species in Cancer Group", x = "Species", y = "Mean Abundance") +
  theme_minimal(base_size = 12)

# Step 10: Display plots
print(p_healthy)
print(p_cancer)

# Optional: Save to file
ggsave("top30_abundant_species_healthy.png", p_healthy, width = 10, height = 8, dpi = 300)
ggsave("top30_abundant_species_cancer.png", p_cancer, width = 10, height = 8, dpi = 300)
```

Phylum Bar plot:

```
# Load libraries
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(viridis)
library(pheatmap)

# Step 1: Read data (phylum in rows, samples in columns)
hh_url <-
"https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/HH_combined_bracken_phylum_fraction.csv"
pc_url <-
"https://raw.githubusercontent.com/okpecallistus/Uchenna5202stuff/refs/heads/main/PC_combined_bracken_phylum_fraction.csv"

hh_phylum <- read.csv(hh_url, row.names = 1, check.names = FALSE)
pC_phylum <- read.csv(pc_url, row.names = 1, check.names = FALSE)

# Step 2: Transpose and reshape to long format
hh_long <- hh_phylum %>%
  t() %>%
  as.data.frame() %>%
  mutate(Sample = rownames(.), Group = "Healthy") %>%
  pivot_longer(-c(Sample, Group), names_to = "Phylum", values_to = "Abundance")

pc_long <- pC_phylum %>%
  t() %>%
  as.data.frame() %>%
  mutate(Sample = rownames(.), Group = "Cancer") %>%
  pivot_longer(-c(Sample, Group), names_to = "Phylum", values_to = "Abundance")

# Step 3: Combine datasets
long_data <- bind_rows(hh_long, pc_long)

# ------------------------------
# ✅ Option 1: Stacked Bar Plot (Normalized)
long_data_norm <- long_data %>%
  group_by(Sample) %>%
  mutate(Abundance = Abundance / sum(Abundance, na.rm = TRUE)) %>%
  ungroup()
```

Bar Plot:



Code:

```
bar_plot<-ggplot(long_data_norm, aes(x = Sample, y = Abundance, fill = Phylum)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ Group, scales = "free_x") +
  theme_minimal(base_size = 12) +
  labs(title = "Gut Microbiome Composition at Phylum Level",
       y = "Relative Abundance", x = "Sample") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  scale_fill_viridis_d(option = "D")
print(bar_plot)
ggsave("phylum_barplot.png", bar_plot, width = 10, height = 6, dpi = 300)
```

Heatmap:

Use Heatmap to show Relative abundance based on Phylum