Fecal Sampling Comparison Project Markdown

Dan Hudson and Xochitl Morgan

```
library(RColorBrewer)
library(Rmisc)
library(vegan)
library(dplyr)
library(phyloseq)
library(ggplot2)
library(ggpubr)
library(DESeq2)

theme_set(theme_bw())
set.seed(123)
```

Load phyloseq object

Phyloseq object was generated on the server using serverScript.R, following the running of this script it was downloaded to the local machine and used to make plots

```
# load data
ps0  <- readRDS("../PrimaryData/phyloObject.rds")

# read metadata
meta <- read.csv("../PrimaryData/purcell_meta.csv")

# add tree
tree <- read_tree("../PrimaryData/purcellFastTree_edit.tre")

# load metadata and tree into phyloseq object
meta <- sample_data(meta)
meta$Individual <- as.factor(meta$Individual)
row.names(meta) <- meta$Sample_name
ps <- merge_phyloseq(ps0, meta, tree)


# unedited phyloseq object
psOG <- ps

# Assign DNA sequences to refseq slot and replace with simple names to improve readability
dna <- Biostrings::DNAStringSet(taxa_names(ps))
names(dna) <- taxa_names(ps)
ps <- merge_phyloseq(ps, dna)
taxa_names(ps) <- paste0("ASV", seq(ntaxa(ps)))
ps
```

```
## phyloseq-class experiment-level object
```

```
## otu_table()   OTU Table:         [ 4872 taxa and 60 samples ]
## sample_data() Sample Data:       [ 60 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 4872 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 4872 tips and 4870 internal nodes ]
## refseq()      DNAStringSet:      [ 4872 reference sequences ]
```

Custom Rarefaction Plot

Not run in this Markdown

```
# Data
psdata <- ps

# Loading required library and displaying core configuration
library(doParallel)
detectCores(all.tests = TRUE)

# Setting up and registering the cluster
cl <- makeCluster(detectCores(all.tests = TRUE)-1)
registerDoParallel(cl)

# Calculate alpha diversity
calculate_rarefaction_curves <- function(psdata, measures, depths, parallel = FALSE) {
  require(plyr) # ldply
  require(reshape2) # melt
  require(doParallel)

  # set parallel options if required
  if (parallel) {
    paropts <- list(.packages = c("phyloseq", "reshape2"))
  } else {
    paropts <- NULL
  }

  estimate_rarified_richness <- function(psdata, measures, depth) {
    if(max(sample_sums(psdata)) < depth) return()
    psdata <- prune_samples(sample_sums(psdata) >= depth, psdata)

    rarified_psdata <- rarefy_even_depth(psdata, depth, verbose = FALSE)

    alpha_diversity <- estimate_richness(rarified_psdata, measures = measures)

    # as.matrix forces the use of melt.array, which includes the Sample names (rownames)
    molten_alpha_diversity <- melt(as.matrix(alpha_diversity),
                                   varnames = c('Sample', 'Measure'),
                                   value.name = 'Alpha_diversity')

    molten_alpha_diversity
  }

  names(depths) <- depths # this enables automatic addition of the Depth to the output by ldply
  rarefaction_curve_data <- ldply(depths,
                                  estimate_rarified_richness,
                                  psdata = psdata,
```

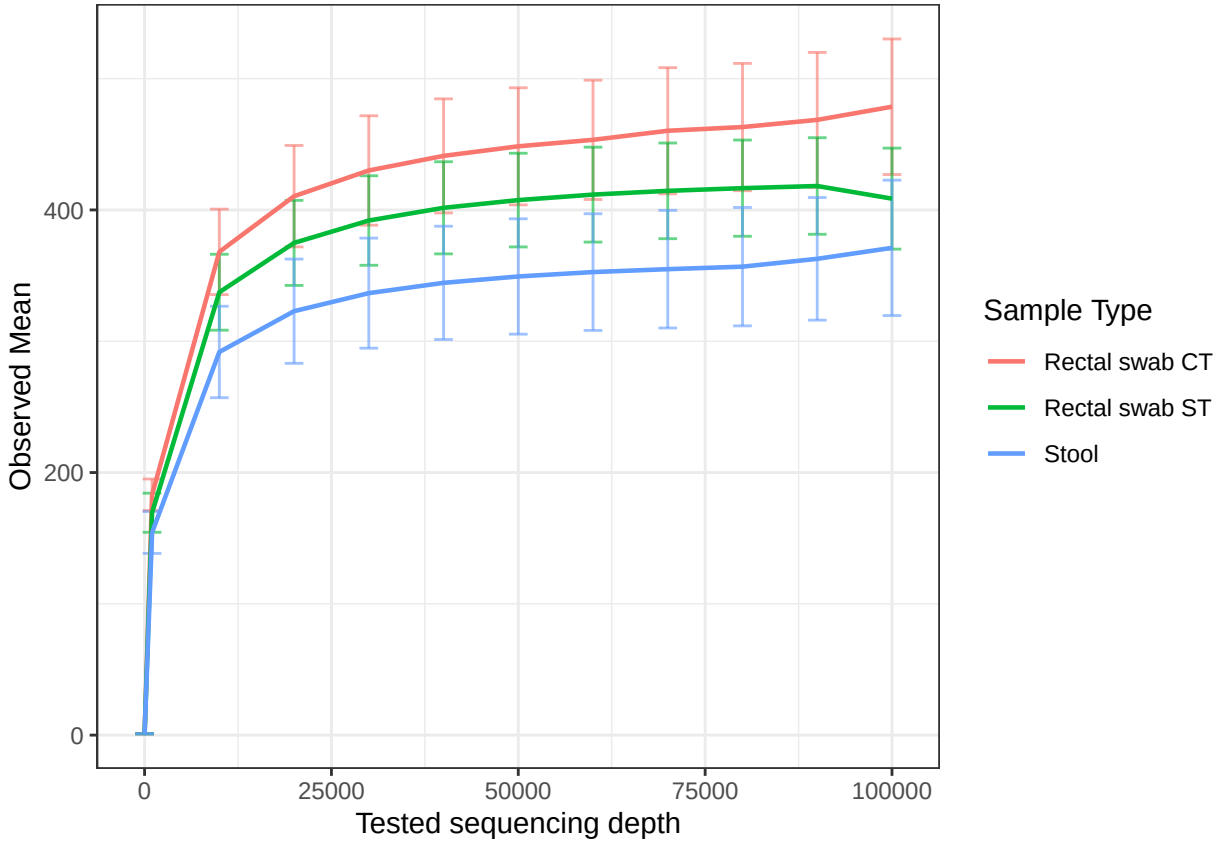
```
                                    measures = measures,
                                    .id = 'Depth',
                                    .progress = ifelse(interactive() && ! parallel, 'text', 'none'),
                                    .parallel = parallel,
                                    .paropts = paropts)

  # convert Depth from factor to numeric
  rarefaction_curve_data$Depth <- as.numeric(levels(rarefaction_curve_data$Depth))[rarefaction_curve_da

  rarefaction_curve_data
}

rarefaction_curve_data <- calculate_rarefaction_curves(psdata, c('Observed'),
                                                       rep(c(1, 100, 1:100 * 1000),
                                                           each = 100))
summary(rarefaction_curve_data)
saveRDS(rarefaction_curve_data, file = "../PrimaryData/rare_object.rds")
```

```
# Data
psdata <- ps

# Load Rarefaction Curve Data Object
rarefaction_curve_data <-
  readRDS(file = "../PrimaryData/rare_object.rds")
summary(rarefaction_curve_data)
```

```
##      Depth             Sample           Measure       Alpha_diversity
##  Min.   :     1   X10B   : 10200   Observed:602300   Min.   :  1.0
##  1st Qu.: 24000   X10C   : 10200                     1st Qu.:306.0
##  Median : 49000   X11A   : 10200                     Median :391.0
##  Mean   : 48853   X11C   : 10200                     Mean   :378.1
##  3rd Qu.: 74000   X12B   : 10200                     3rd Qu.:451.0
##  Max.   :100000   X12C   : 10200                     Max.   :672.0
##                   (Other):541100
```

```
# Summarise alpha diversity
rarefaction_curve_data_summary <-
  ddply(rarefaction_curve_data,
        c('Depth', 'Sample', 'Measure'),
        summarise,
        Alpha_diversity_mean = mean(Alpha_diversity),
        Alpha_diversity_sd = sd(Alpha_diversity))

colnames(rarefaction_curve_data_summary) <-
  gsub("X","", colnames(rarefaction_curve_data_summary))
rarefaction_curve_data_summary$Sample <-
  gsub("X","", rarefaction_curve_data_summary$Sample)

# Add sample data
rarefaction_curve_data_summary_verbose <-
  merge(rarefaction_curve_data_summary,
        data.frame(sample_data(psdata)),
        by.x = 'Sample',
        by.y = 'row.names')
```

```
# Produce summary df of rarefaction data
df_mod <- summarySE(rarefaction_curve_data_summary_verbose,
                    measurevar = "Alpha_diversity_mean",
                    groupvars = c("Depth", "Sample_type"))
df_mod <- df_mod %>%
  subset(Depth == 1 | Depth == 1000 | Depth == 10000 | Depth == 20000 |
           Depth == 30000 | Depth == 40000 | Depth == 50000 | Depth == 60000 |
           Depth == 70000 | Depth == 80000 | Depth == 90000 | Depth == 100000)

ggplot(df_mod, aes(x = Depth,
                   y = Alpha_diversity_mean,
                   ymin = Alpha_diversity_mean - ci,
                   ymax = Alpha_diversity_mean + ci,
                   colour = Sample_type)) +
  geom_errorbar(size = 0.5, width = 2500, alpha = 0.6) +
  geom_line(size = 0.8) +
  labs(x = "Tested sequencing depth",
       y = "Observed Mean", color = "Sample Type")
```




```
ggsave("../Results/S1)Rarefaction_Curve.pdf", width = 11, height = 8)
```

Rarefy

```
# Rarefy to even sequencing depth, 90% of minimum sample depth, seed for randomness is 1
ps_rare <- rarefy_even_depth(ps, rngseed = 1,
```

```
                                sample.size = 0.9 * min(sample_sums(ps)),
                                replace = FALSE)
```

```
## `set.seed(1)` was used to initialize repeatable random subsampling.

## Please record this for your records so others can reproduce.

## Try `set.seed(1); .Random.seed` for the full vector

## ...

## 2430TUs were removed because they are no longer
## present in any sample after random subsampling

## ...
```

```
sample_sums(ps)
```

```
##    10A    10B    10C    11A    11B    11C    12A    12B    12C    13A    13B
##  97672 152224 136830 107226  92295 142349  63696 151049 153224 170086 154765
##    13C    14A    14B    14C    15A    15B    15C    16A    16B    16C    17A
## 146933 160605 171722 140943 175324 114245 168613 120816 131462 141789 153959
##    17B    17C    18A    18B    18C    19A    19B    19C     1A     1B     1C
## 127615  94965 160212 126836 159814 161407 153370 121330 165497  96844 113268
##    20A    20B    20C     2A     2B     2C     3A     3B     3C     4A     4B
## 195853 115506 127239 110007 118680 110327 146390 136636 106307 104581 125868
##     4C     5A     5B     5C     6A     6B     6C     7A     7B     7C     8A
## 131775 160742 121440  88650 140459 164106  92481 137767 138331 120381 140622
##     8B     8C     9A     9B     9C
##  97857 112182  84876 143122 108117
```

```
sample_sums(ps_rare)
```

```
##   10A   10B   10C   11A   11B   11C   12A   12B   12C   13A   13B   13C   14A
## 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326
##   14B   14C   15A   15B   15C   16A   16B   16C   17A   17B   17C   18A   18B
## 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326
##   18C   19A   19B   19C    1A    1B    1C   20A   20B   20C    2A    2B    2C
## 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326
##    3A    3B    3C    4A    4B    4C    5A    5B    5C    6A    6B    6C    7A
## 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326 57326
##    7B    7C    8A    8B    8C    9A    9B    9C
## 57326 57326 57326 57326 57326 57326 57326 57326
```

Alpha Diversity

```
# Calculate alpha diversity, using Richness and Shannon
alpha_summary <- estimate_richness(ps_rare, measures = c("Observed", "Shannon"))

# Use Shapiro-Wilk normality test to determine the test to be used
shapiro.test(alpha_summary$Observed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha_summary$Observed
## W = 0.99288, p-value = 0.9797
```

```r
shapiro.test(alpha_summary$Shannon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha_summary$Shannon
## W = 0.97844, p-value = 0.3662
```

```r
# Blocking Test
rO <- alpha_summary$Observed
rS <- alpha_summary$Shannon

f <- c("Clinician", "Self", "Stool") # treatment levels
k <- 3 # number of treatment levels
n <- 20 # number of control blocks

tm <- gl(k, 1, n*k, factor(f)) # matching treatment
blk <- gl(n, k, k*n) # blocking factor

avO <- aov(rO ~ tm + blk)
summary(avO)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## tm           2 106186   53093  14.357 2.27e-05 ***
## blk         19 332244   17487   4.729 2.33e-05 ***
## Residuals   38 140525    3698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
avS <- aov(rS ~ tm + blk)
summary(avS)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## tm           2  0.849  0.4247   6.675 0.003278 **
## blk         19  4.871  0.2564   4.030 0.000126 ***
## Residuals   38  2.418  0.0636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Test whether the observed number of OTUs differs significantly between samples
# p adjustment using Benjamini and Hochberg
pairwise.t.test(alpha_summary$Observed, sample_data(ps_rare)$Sample_type, p.adjust = "BH")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  alpha_summary$Observed and sample_data(ps_rare)$Sample_type
##
##                Rectal swab CT Rectal swab ST
## Rectal swab ST 0.1409         -
## Stool          0.0023         0.0646
##
## P value adjustment method: BH
```

```r
pairwise.t.test(alpha_summary$Shannon, sample_data(ps_rare)$Sample_type, p.adjust = "BH")
```

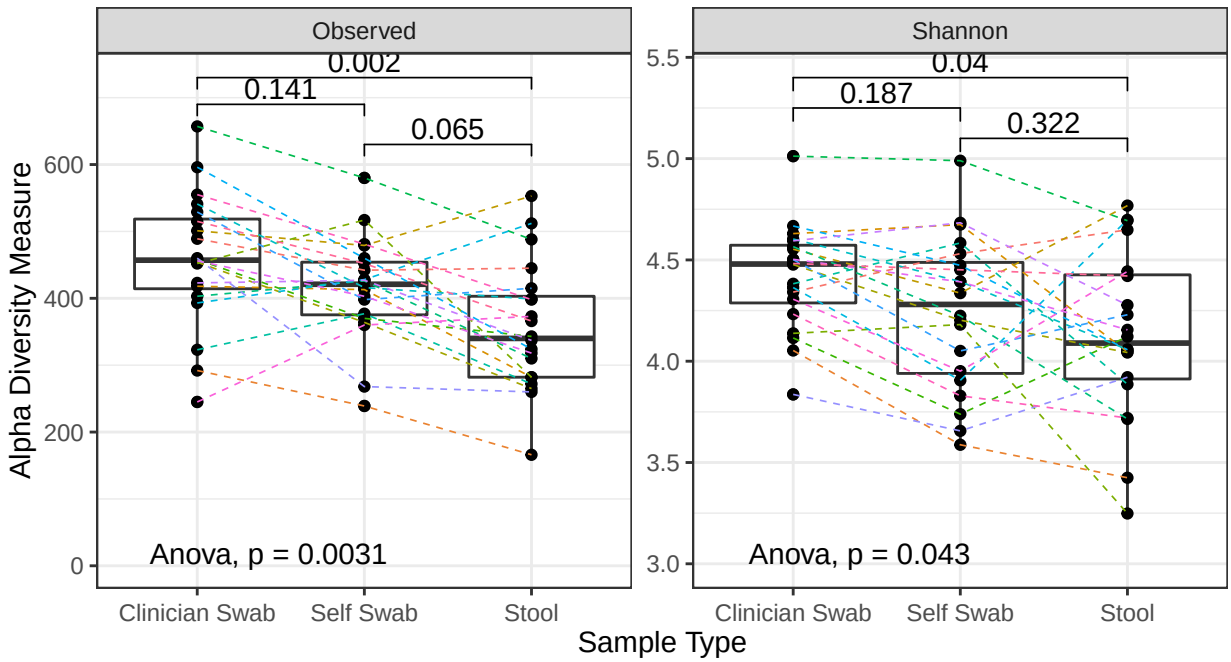
```
##
```

```
##  Pairwise comparisons using t tests with pooled SD
##
## data:  alpha_summary$Shannon and sample_data(ps_rare)$Sample_type
##
##                Rectal swab CT Rectal swab ST
## Rectal swab ST 0.19           -
## Stool          0.04           0.32
##
## P value adjustment method: BH
```

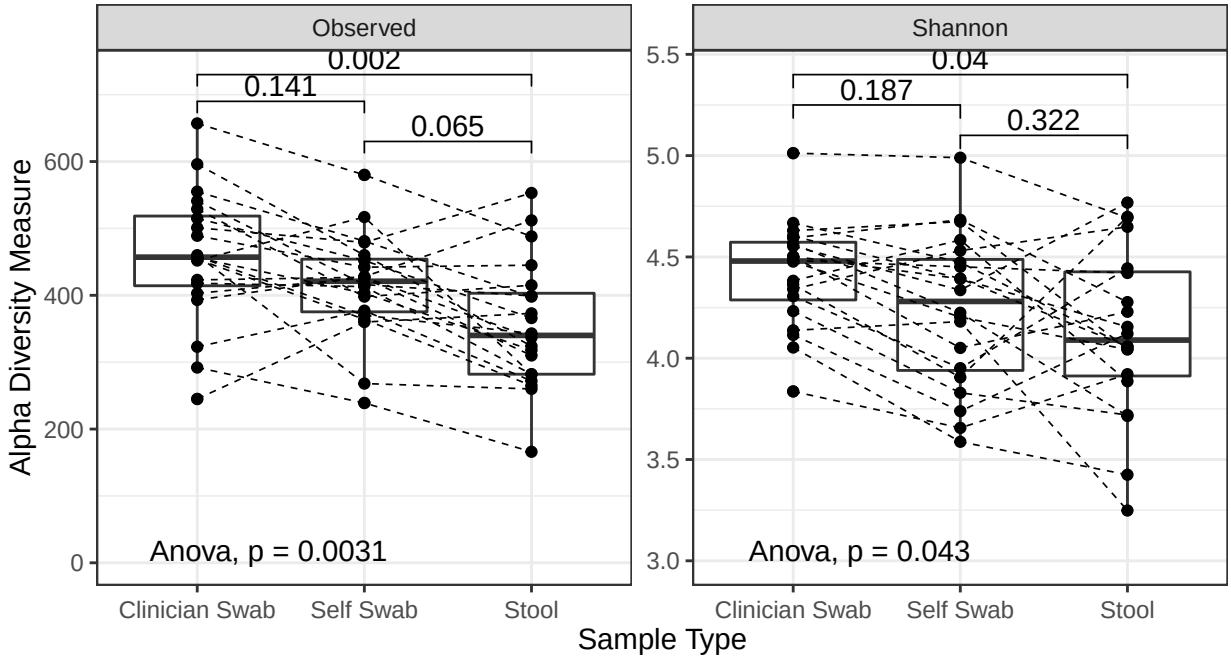
```
# Make adjusted p value dataframe
pObs <- pairwise.t.test(alpha_summary$Observed, sample_data(ps_rare)$Sample_type, p.adjust = "BH")
pSha <- pairwise.t.test(alpha_summary$Shannon, sample_data(ps_rare)$Sample_type, p.adjust = "BH")

variable <- c("Observed", "Observed", "Observed", "Shannon", "Shannon", "Shannon")
group1 <- c("Rectal swab CT", "Rectal swab ST", "Rectal swab CT",
            "Rectal swab CT", "Rectal swab ST", "Rectal swab CT")
group2 <- c("Stool", "Stool", "Rectal swab ST", "Stool", "Stool", "Rectal swab ST")
pVal <- c(round(pObs$p.value[2,1], 3), round(pObs$p.value[2,2], 3), round(pObs$p.value[1,1], 3),
       round(pSha$p.value[2,1], 3), round(pSha$p.value[2,2], 3), round(pSha$p.value[1,1], 3))
y.position <- c(730, 630, 690, 5.4, 5.1, 5.25)

pAdjusted <- bind_cols(variable, group1, group2, pVal, y.position)
```

```
## New names:
## * NA -> ...1
## * NA -> ...2
## * NA -> ...3
## * NA -> ...4
## * NA -> ...5
```

```
colnames(pAdjusted) <- c("variable", "group1", "group2", "p", "y.position")

# Plot Observed richness, Shannon, and Simpson diversity values
p <- plot_richness(ps_rare, x = "Sample_type",
                   measures = c("Observed", "Shannon"))

# Add boxplot, individual data points, and linked lines using geom layers
p$layers <- p$layers[-1]
p + geom_boxplot() + geom_point() + xlab("Sample Type") +
  geom_line(aes(group = Individual, colour = Individual), size = 0.3, linetype = "dashed") +
  scale_x_discrete(labels=c("Rectal swab CT" = "Clinician Swab",
                            "Rectal swab ST" = "Self Swab",
                            "Stool" = "Stool")) +
  theme(axis.text.x = element_text(angle = 360, hjust = 0.5),
        aspect.ratio = 1, legend.position = "none") +
  stat_pvalue_manual(pAdjusted) +
  stat_compare_means(method = "anova", label.y = 3)
```




```
ggsave("../Results/1)Alpha_Diversity_Colour.pdf", width = 7, height = 4)


p + geom_boxplot() + geom_point() + xlab("Sample Type") +
  geom_line(aes(group = Individual), size = 0.3, linetype = "dashed") +
  scale_x_discrete(labels=c("Rectal swab CT" = "Clinician Swab",
                            "Rectal swab ST" = "Self Swab",
                            "Stool" = "Stool")) +
  theme(axis.text.x = element_text(angle = 360, hjust = 0.5),
        aspect.ratio = 1, legend.position = "none") +
  stat_pvalue_manual(pAdjusted) +
  stat_compare_means(method = "anova", label.y = 3)
```




```
ggsave("../Results/1)Alpha_Diversity_Plain.pdf", width = 7, height = 4)
```

Beta Diversity - Bray-Curtis - Using Rarefied Data

```
# Ordinate data using Non-metric multidimensional scaling (NMDS) on Bray-Curtis dissimilarity (distance
bray_dist <- phyloseq::distance(ps_rare, method = "bray")
ord.nmds.bray <- ordinate(ps_rare, "NMDS", "bray")
```

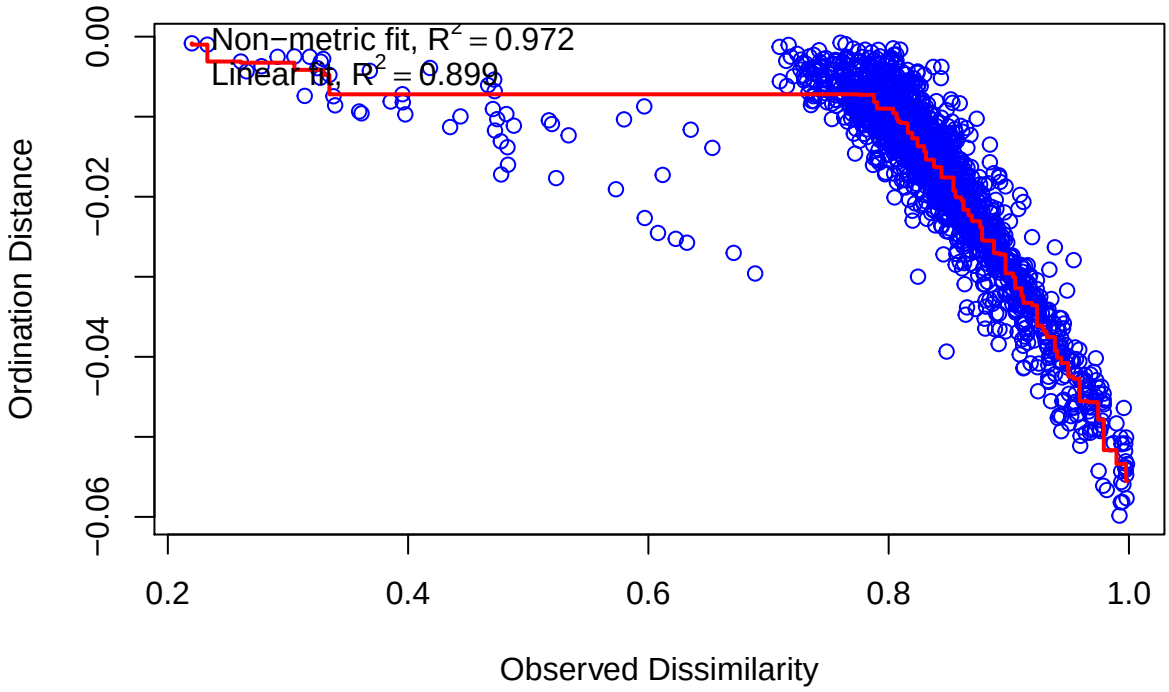
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1711863
## Run 1 stress 0.1684071
## ... New best solution
## ... Procrustes: rmse 0.04413298  max resid 0.1986277
## Run 2 stress 0.1712493
## Run 3 stress 0.171249
## Run 4 stress 0.171249
## Run 5 stress 0.1684069
## ... New best solution
## ... Procrustes: rmse 0.0001460378  max resid 0.0008854301
## ... Similar to previous best
## Run 6 stress 0.1684186
## ... Procrustes: rmse 0.001341072  max resid 0.006882538
## ... Similar to previous best
## Run 7 stress 0.1711861
## Run 8 stress 0.171249
## Run 9 stress 0.1711862
## Run 10 stress 0.1684181
## ... Procrustes: rmse 0.001248294  max resid 0.006799046
## ... Similar to previous best
## Run 11 stress 0.1684069
## ... New best solution
```

```
## ... Procrustes: rmse 5.458079e-05  max resid 0.0003677635
## ... Similar to previous best
## Run 12 stress 0.2132782
## Run 13 stress 0.2033464
## Run 14 stress 0.168407
## ... Procrustes: rmse 6.000605e-05  max resid 0.0003168814
## ... Similar to previous best
## Run 15 stress 0.2049706
## Run 16 stress 0.1684072
## ... Procrustes: rmse 0.0001684513  max resid 0.001050944
## ... Similar to previous best
## Run 17 stress 0.1684069
## ... Procrustes: rmse 1.018318e-05  max resid 4.545695e-05
## ... Similar to previous best
## Run 18 stress 0.1712437
## Run 19 stress 0.168407
## ... Procrustes: rmse 5.314934e-05  max resid 0.0003272289
## ... Similar to previous best
## Run 20 stress 0.1712434
## *** Solution reached
```

```
# Call newly created file to get the stress value of the plot
ord.nmds.bray
```

```
##
## Call:
## metaMDS(comm = veganifyOTU(physeq), distance = distance)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(veganifyOTU(physeq)))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1684069
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(sqrt(veganifyOTU(physeq)))'
```

```
# Stress plot
stressplot(ord.nmds.bray)
```




```
# Stats
# Test whether the sample types differ significantly from each other using PERMANOVA
adonis(bray_dist ~ sample_data(ps_rare)$Sample_type)
```

```
##
## Call:
## adonis(formula = bray_dist ~ sample_data(ps_rare)$Sample_type)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                  Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## sample_data(ps_rare)$Sample_type  2    1.3629 0.68145  2.0786 0.06798  0.001
## Residuals                        57   18.6866 0.32784         0.93202
## Total                            59   20.0495                 1.00000
##
## sample_data(ps_rare)$Sample_type ***
## Residuals
## Total
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(formula = bray_dist ~ sample_data(ps_rare)$Individual)
```

```
##
## Call:
## adonis(formula = bray_dist ~ sample_data(ps_rare)$Individual)
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Terms added sequentially (first to last)
##
##                                Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
## sample_data(ps_rare)$Individual 19   14.9269 0.78563  6.1346 0.7445  0.001 ***
## Residuals                       40    5.1226 0.12807         0.2555
## Total                           59   20.0495                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(bray_dist ~ sample_data(ps_rare)$Sample_type*sample_data(ps_rare)$Individual)
```

```
##
## Call:
## adonis(formula = bray_dist ~ sample_data(ps_rare)$Sample_type *      sample_data(ps_rare)$Individual
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                                                 Df SumsOfSqs
## sample_data(ps_rare)$Sample_type                                 2    1.3629
## sample_data(ps_rare)$Individual                                 19   14.9269
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 38    3.7597
## Residuals                                                        0    0.0000
## Total                                                           59   20.0495
##                                                                 MeanSqs
## sample_data(ps_rare)$Sample_type                                      1
## sample_data(ps_rare)$Individual                                       1
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual      0
## Residuals                                                           Inf
## Total
##                                                                 F.Model
## sample_data(ps_rare)$Sample_type                                      0
## sample_data(ps_rare)$Individual                                       0
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual      0
## Residuals
## Total
##                                                                      R2 Pr(>F)
## sample_data(ps_rare)$Sample_type                                0.06798      1
## sample_data(ps_rare)$Individual                                 0.74450      1
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 0.18752      1
## Residuals                                                       0.00000
## Total                                                           1.00000
```

```
anosim(bray_dist, sample_data(ps_rare)$Sample_type)
```

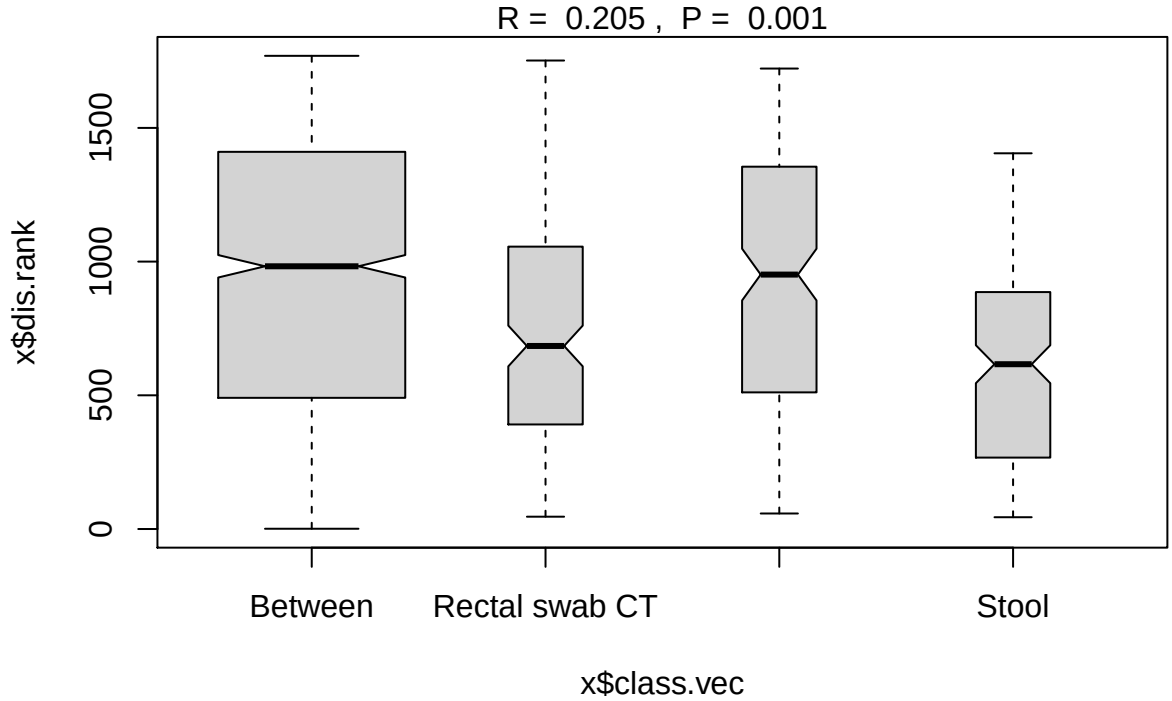
```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.2048
##       Significance: 0.001
##
```

```
## Permutation: free
## Number of permutations: 999
```

```
BCanoSamp <- (anosim(bray_dist, sample_data(ps_rare)$Sample_type))
summary(BCanoSamp)
```

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.2048
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0296 0.0435 0.0568 0.0732
##
## Dissimilarity ranks between and within classes:
##                 0%     25%    50%     75%   100%    N
## Between          1 491.250 982.50 1410.50 1769.5 1200
## Rectal swab CT  46 392.750 684.50 1053.25 1752.0  190
## Rectal swab ST  58 513.125 951.50 1353.25 1722.0  190
## Stool           44 267.750 616.25  885.75 1405.0  190
```

```
plot(BCanoSamp)
```




```
anosim(bray_dist, sample_data(ps_rare)$Individual)
```

```
##
```

```
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.789
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

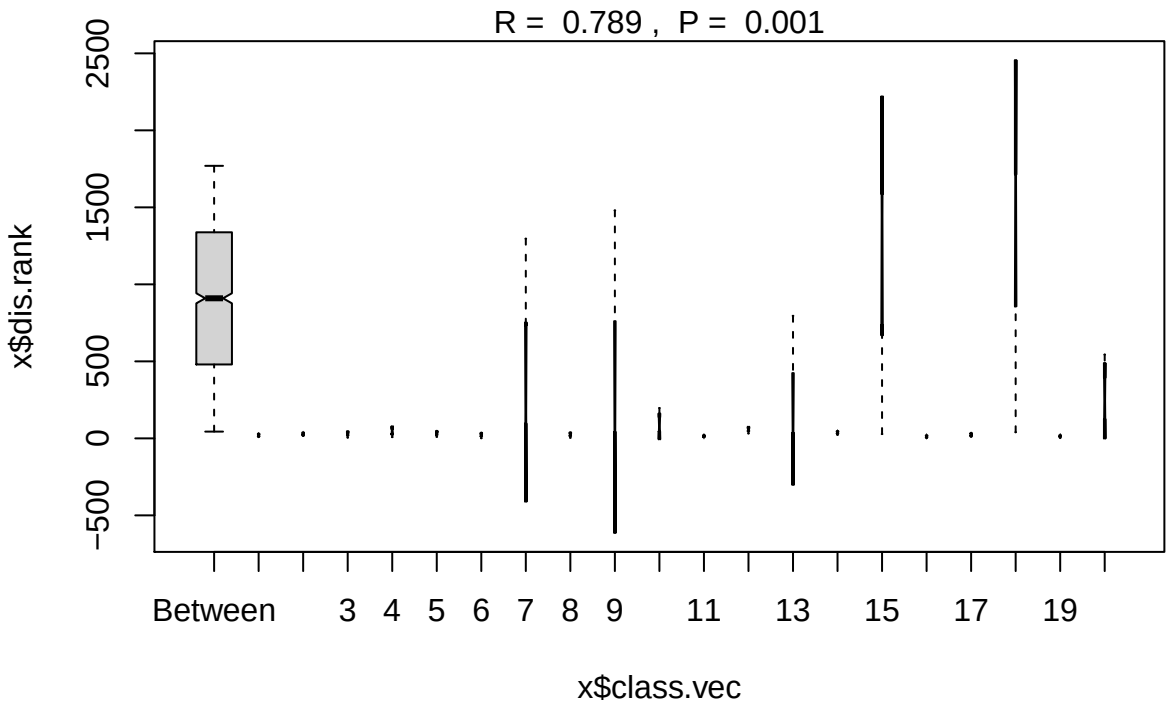
```
BCanoInd <- anosim(bray_dist, sample_data(ps_rare)$Individual)
summary(BCanoInd)
```

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.789
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0732 0.1019 0.1270 0.1512
##
## Dissimilarity ranks between and within classes:
##         0%    25%    50%      75%   100%    N
## Between 44 480.25  909.5 1337.625 1769.5 1710
## 1        9  15.00   21.0   25.500   30.0    3
## 2       16  22.50   29.0   33.000   37.0    3
## 3        5  19.00   33.0   34.000   35.0    3
## 4        8  29.50   51.0   60.000   69.0    3
## 5       10  23.00   36.0   37.000   38.0    3
## 6        1  13.00   25.0   25.500   26.0    3
## 7       20  95.50  171.0  733.750 1296.5    3
## 8        4  15.50   27.0   29.000   31.0    3
## 9       41  41.50   42.0  761.000 1480.0    3
## 10      12  45.00   78.0  137.500  197.0    3
## 11       7  10.50   14.0   18.000   22.0    3
## 12      32  46.50   61.0   62.500   64.0    3
## 13      19  35.50   52.0  423.750  795.5    3
## 14      23  31.00   39.0   41.000   43.0    3
## 15      28 736.50 1445.0 1587.000 1729.0    3
## 16       2   7.50   13.0   15.500   18.0    3
## 17      11  17.50   24.0   29.000   34.0    3
## 18      40 857.00 1674.0 1714.500 1755.0    3
## 19       3   9.00   15.0   16.000   17.0    3
## 20       6 125.00  244.0  394.000  544.0    3
```

```
plot(BCanoInd)
```

```
## Warning in bxp(list(stats = structure(c(44, 480, 909.5, 1338, 1769.5, 9, : some
```

```
## notches went outside hinges ('box'): maybe set notch=FALSE
```




```
BCps.disper <- betadisper(bray_dist, sample_data(ps_rare)$Sample_type)
anova(BCps.disper)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df   Sum Sq   Mean Sq F value  Pr(>F)
## Groups     2 0.010371 0.0051855  2.6804 0.07717 .
## Residuals 57 0.110273 0.0019346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(BCps.disper)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.010371 0.0051855 2.6804    999  0.087 .
## Residuals 57 0.110273 0.0019346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(BCps.disper, pairwise = TRUE)
```

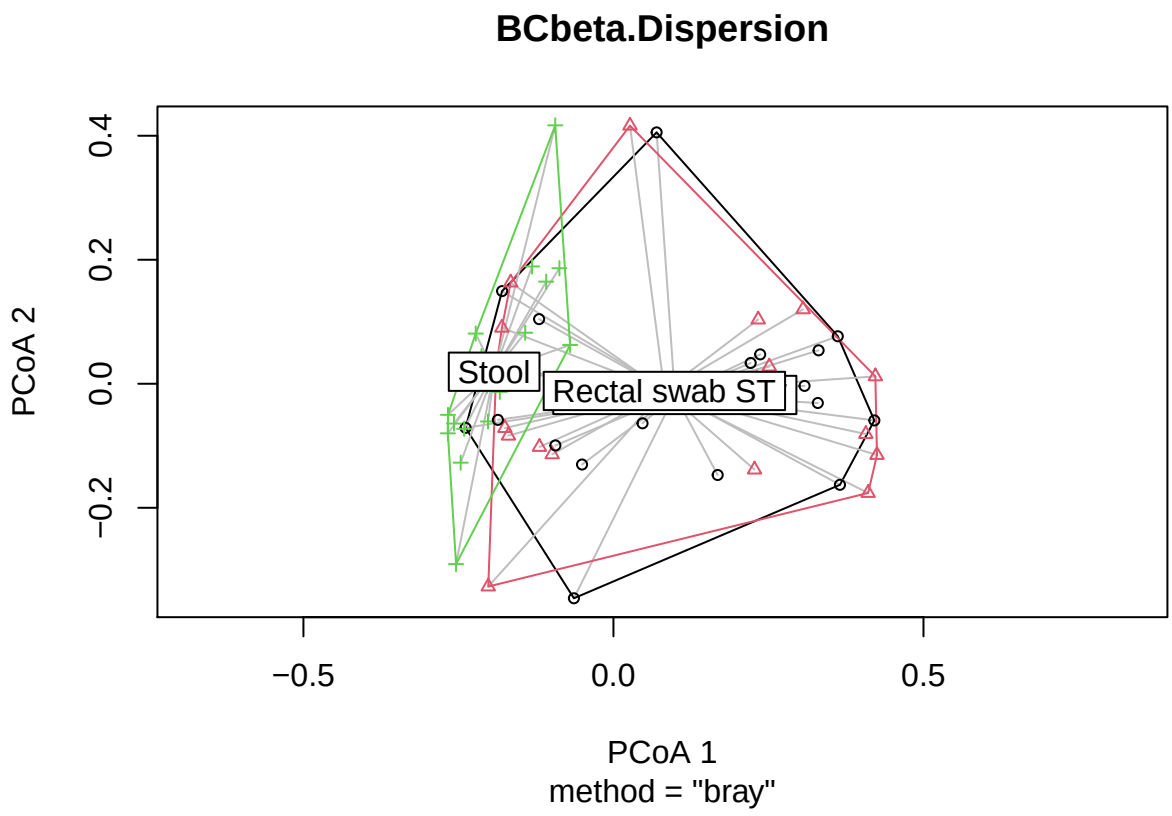
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
```

```
##
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.010371 0.0051855 2.6804    999  0.087 .
## Residuals 57 0.110273 0.0019346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                Rectal swab CT Rectal swab ST Stool
## Rectal swab CT                       0.152000 0.402
## Rectal swab ST       0.156123                 0.034
## Stool                0.395585       0.028482
```

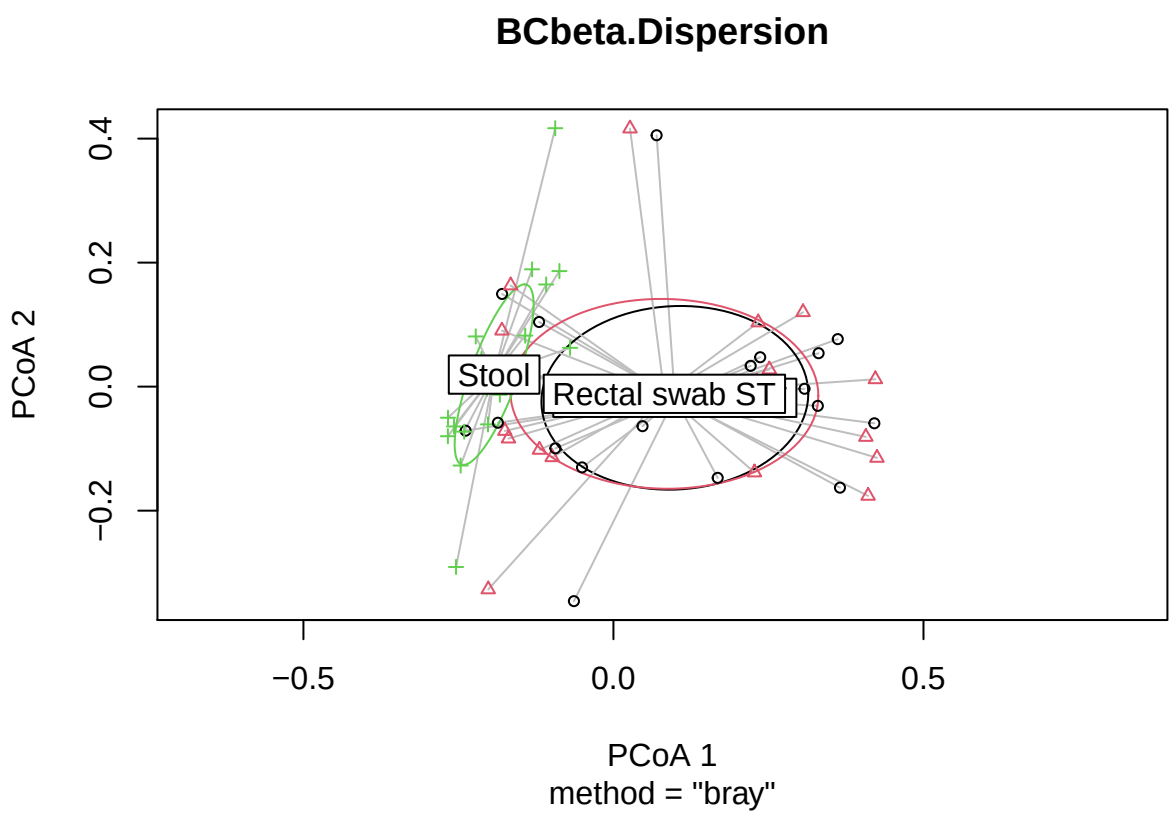
```
TukeyHSD(BCps.disper)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                     diff         lwr        upr     p adj
## Rectal swab ST-Rectal swab CT  0.01985443 -0.01361660 0.05332546 0.3337714
## Stool-Rectal swab CT          -0.01203124 -0.04550227 0.02143979 0.6643777
## Stool-Rectal swab ST          -0.03188567 -0.06535670 0.00158536 0.0649792
```

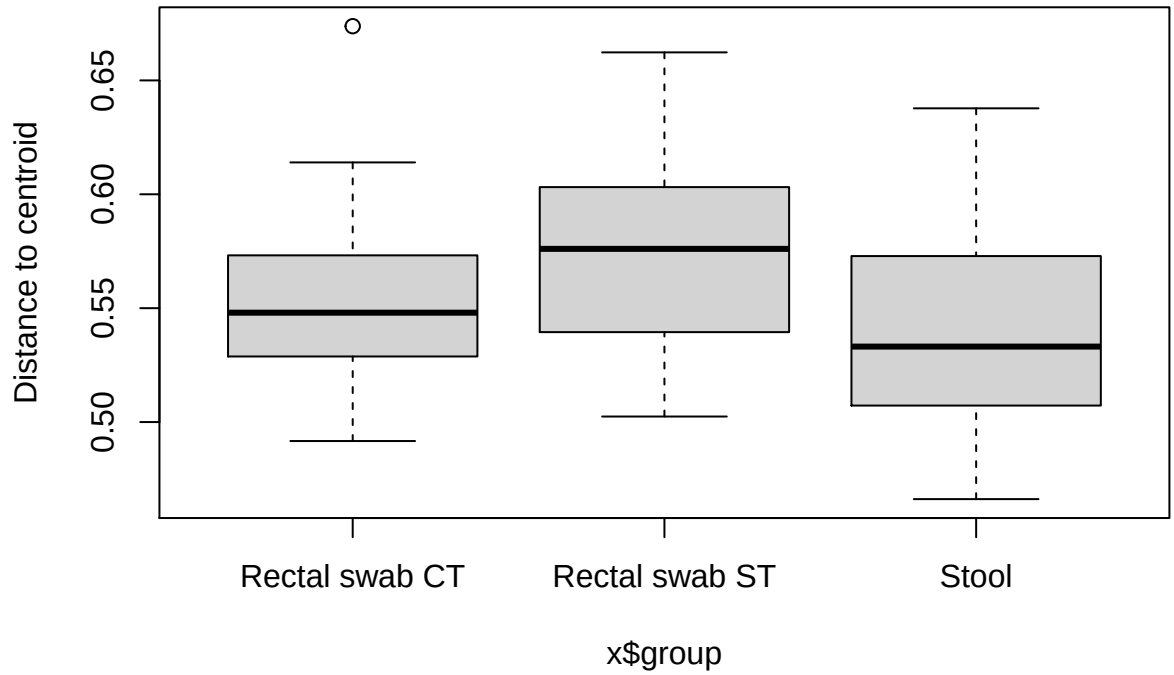
```
# Beta Dispersion Plots
BCbeta.Dispersion <- BCps.disper
plot(BCbeta.Dispersion)
```



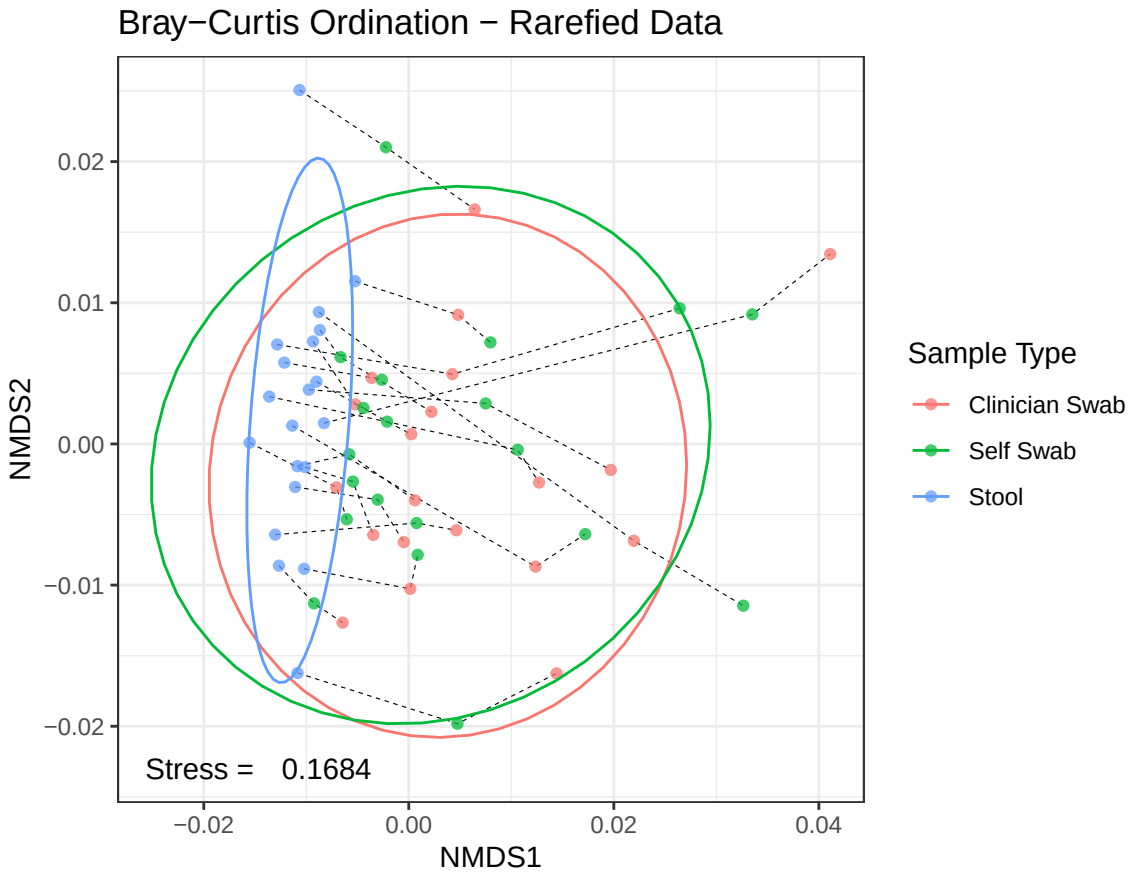

```
plot(BCbeta.Dispersion, hull = FALSE, ellipse = TRUE)
```



```
boxplot(BCbeta.Dispersion)
```




```
# Bray-Curtis NMDS plot
bcdf <- plot_ordination(ps_rare, ord.nmds.bray, justDF = TRUE)
bcdf$Sample_type <- gsub("Rectal swab CT", "Clinician Swab", bcdf$Sample_type)
bcdf$Sample_type <- gsub("Rectal swab ST", "Self Swab", bcdf$Sample_type)

BC_plot <- ggplot(bcdf, aes(x = NMDS1, y = NMDS2)) +
  geom_line(aes(group = Individual), size = 0.2, linetype = "dashed") +
  geom_point(aes(color = Sample_type), alpha = 0.75) +
  stat_ellipse(aes(color = Sample_type)) +
  annotate("text", x = -0.02, y = -0.023, label = "Stress = ") +
  annotate("text", x = -0.008, y = -0.023, label = round(ord.nmds.bray$stress, 4)) +
  labs(title = "Bray-Curtis Ordination - Rarefied Data", color = "Sample Type") +
  theme(aspect.ratio = 1, plot.margin = unit(c(0, 1, 0, 0), "pt"))

BC_plot
```




```
ggsave("../Results/2A)Beta_Diversity_BC_rare.pdf", width = 6, height = 4.5)
```

Beta Diversity - Bray-Curtis - Using Unrarefied Data

```
# Ordinate data using Non-metric multidimensional scaling (NMDS) on Bray-Curtis dissimilarity (distance
bray_distNR <- phyloseq::distance(ps, method = "bray")
ord.nmds.brayNR <- ordinate(ps, "NMDS", "bray")
```

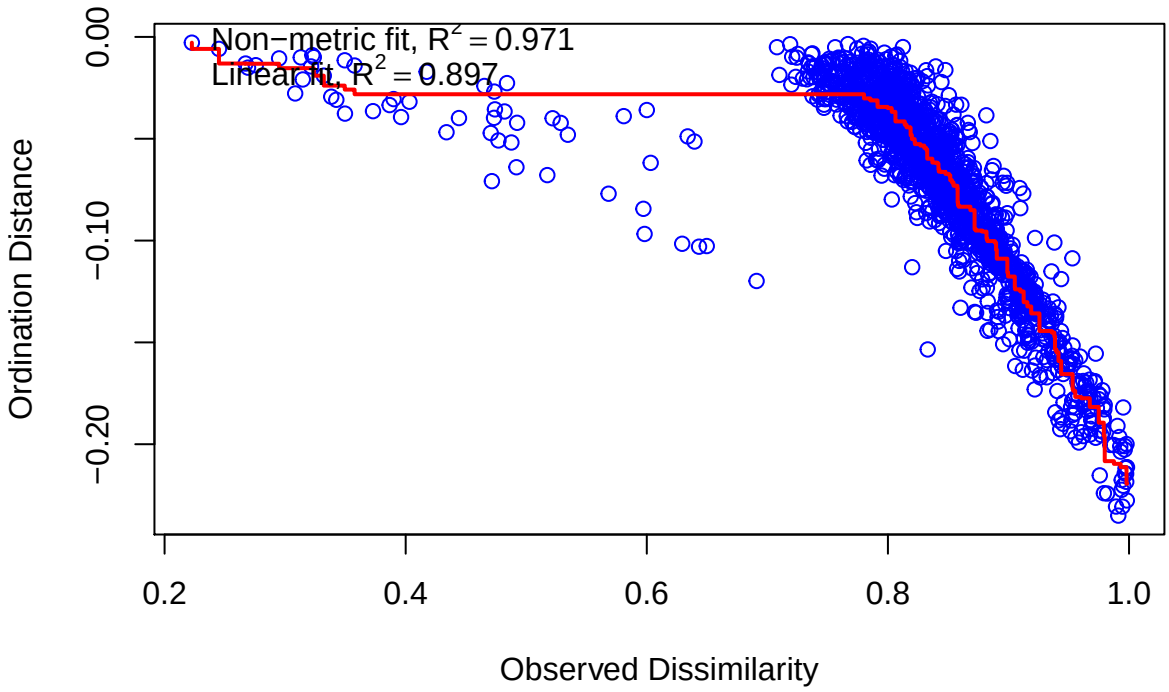
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1727247
## Run 1 stress 0.1735502
## Run 2 stress 0.1699237
## ... New best solution
## ... Procrustes: rmse 0.04091108  max resid 0.1915145
## Run 3 stress 0.1727247
## Run 4 stress 0.169934
## ... Procrustes: rmse 0.001703474  max resid 0.01032407
## Run 5 stress 0.2038782
## Run 6 stress 0.1785011
## Run 7 stress 0.1731887
## Run 8 stress 0.1728522
## Run 9 stress 0.1786149
## Run 10 stress 0.1832383
## Run 11 stress 0.1702377
```

```
## ... Procrustes: rmse 0.006672044  max resid 0.0414537
## Run 12 stress 0.1702376
## ... Procrustes: rmse 0.006670562  max resid 0.04139859
## Run 13 stress 0.1702376
## ... Procrustes: rmse 0.006671683  max resid 0.04141045
## Run 14 stress 0.1735499
## Run 15 stress 0.1827916
## Run 16 stress 0.1699237
## ... New best solution
## ... Procrustes: rmse 2.777038e-05  max resid 0.0001633478
## ... Similar to previous best
## Run 17 stress 0.1831288
## Run 18 stress 0.1732153
## Run 19 stress 0.1702377
## ... Procrustes: rmse 0.006673308  max resid 0.04139626
## Run 20 stress 0.1832185
## *** Solution reached
```

```
# Call newly created file to get the stress value of the plot
ord.nmds.brayNR
```

```
##
## Call:
## metaMDS(comm = veganifyOTU(physeq), distance = distance)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(veganifyOTU(physeq)))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1699237
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(sqrt(veganifyOTU(physeq)))'
```

```
# Stress plot
stressplot(ord.nmds.brayNR)
```




```
# Stats
# Test whether the sample types differ significantly from each other using PERMANOVA
adonis(bray_distNR ~ sample_data(ps)$Sample_type)
```

```
##
## Call:
## adonis(formula = bray_distNR ~ sample_data(ps)$Sample_type)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                             Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## sample_data(ps)$Sample_type  2     1.306 0.65300  1.9753 0.06482  0.002 **
## Residuals                   57    18.843 0.33058         0.93518
## Total                       59    20.149                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(formula = bray_distNR ~ sample_data(ps)$Individual)
```

```
##
## Call:
## adonis(formula = bray_distNR ~ sample_data(ps)$Individual)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## sample_data(ps)$Individual 19   14.8936 0.78387   5.966 0.73916  0.001 ***
```

```
## Residuals                  40    5.2556 0.13139         0.26084
## Total                      59   20.1492                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(bray_distNR ~ sample_data(ps)$Sample_type*sample_data(ps)$Individual)
```

```
##
## Call:
## adonis(formula = bray_distNR ~ sample_data(ps)$Sample_type *      sample_data(ps)$Individual)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                                          Df SumsOfSqs MeanSqs
## sample_data(ps)$Sample_type                               2    1.3060       1
## sample_data(ps)$Individual                               19   14.8936       1
## sample_data(ps)$Sample_type:sample_data(ps)$Individual 38    3.9496       0
## Residuals                                                 0    0.0000     Inf
## Total                                                    59   20.1492
##                                                          F.Model      R2 Pr(>F)
## sample_data(ps)$Sample_type                                    0 0.06482      1
## sample_data(ps)$Individual                                     0 0.73916      1
## sample_data(ps)$Sample_type:sample_data(ps)$Individual         0 0.19602      1
## Residuals                                                        0.00000
## Total                                                            1.00000
```

```
anosim(bray_distNR, sample_data(ps)$Sample_type)
```

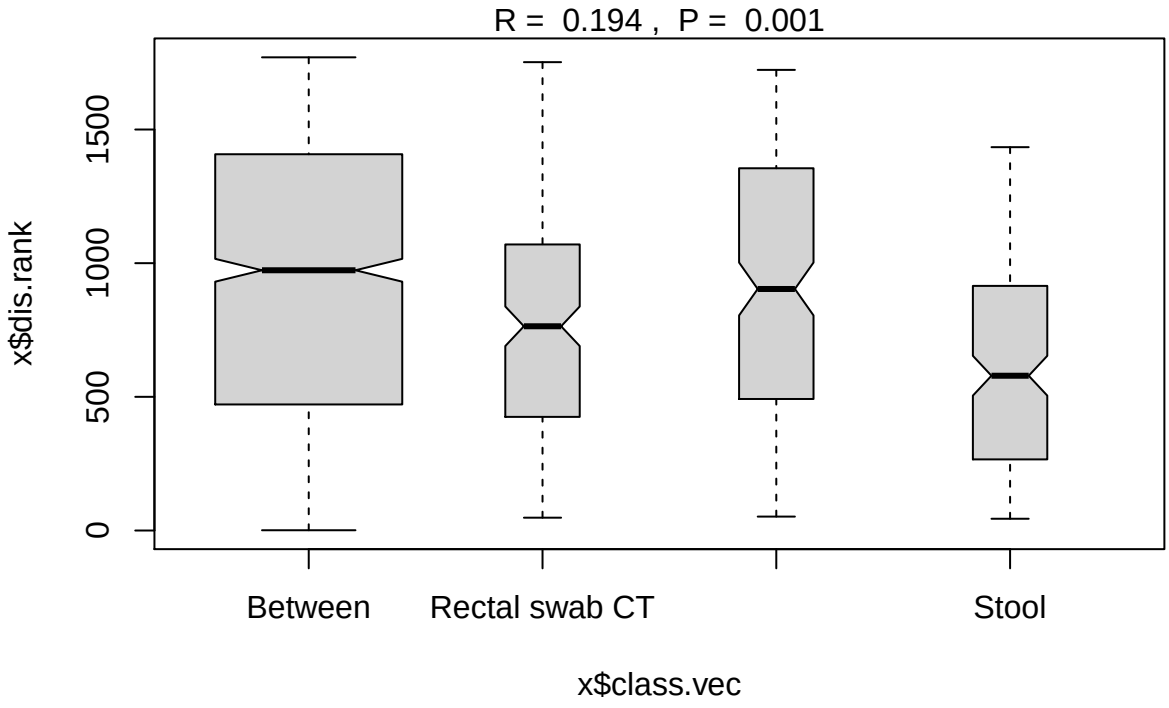
```
##
## Call:
## anosim(x = bray_distNR, grouping = sample_data(ps)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.1939
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

```
BCanoSampNR <- (anosim(bray_distNR, sample_data(ps)$Sample_type))
summary(BCanoSampNR)
```

```
##
## Call:
## anosim(x = bray_distNR, grouping = sample_data(ps)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.1939
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0315 0.0434 0.0563 0.0715
##
## Dissimilarity ranks between and within classes:
##                0%    25%   50%     75% 100%    N
## Between         1 471.75 973.5 1407.25 1770 1200
## Rectal swab CT 48 427.00 764.0 1066.25 1752  190
## Rectal swab ST 52 493.50 903.5 1354.25 1723  190
## Stool          44 266.25 579.0  914.75 1434  190
```

```
plot(BCanoSampNR)
```




```
anosim(bray_distNR, sample_data(ps)$Individual)
```

```
##
## Call:
## anosim(x = bray_distNR, grouping = sample_data(ps)$Individual)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.7908
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

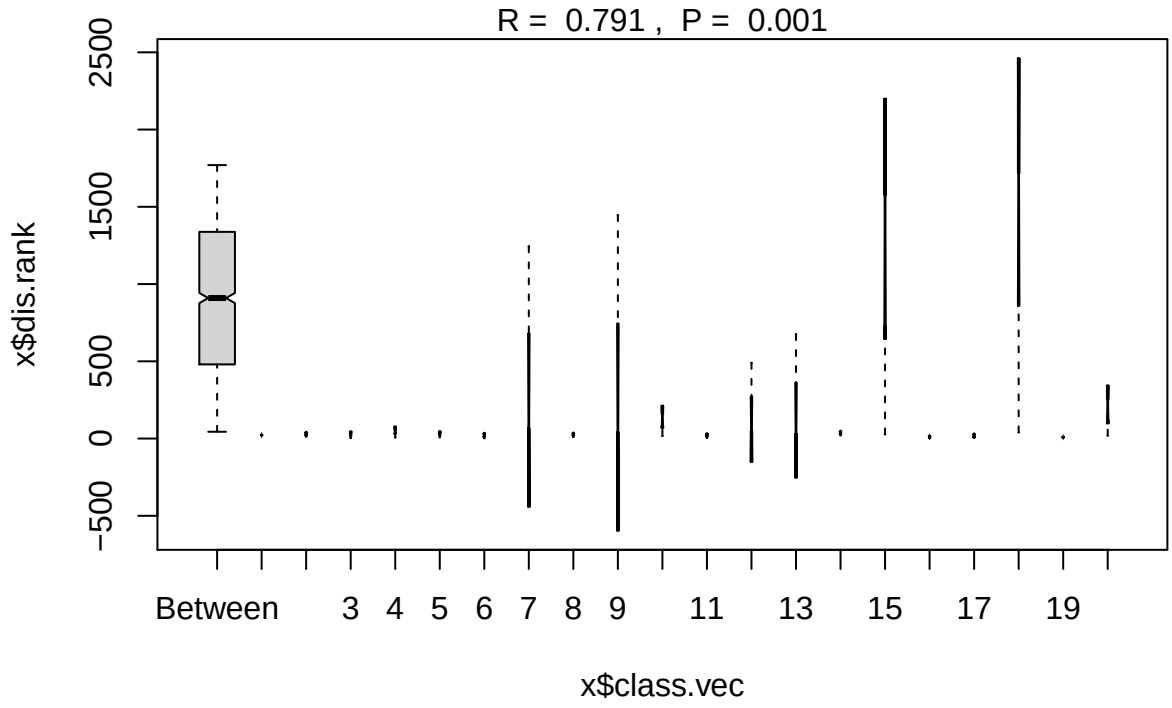
```
BCanoIndNR <- anosim(bray_distNR, sample_data(ps)$Individual)
summary(BCanoIndNR)
```

```
##
## Call:
## anosim(x = bray_distNR, grouping = sample_data(ps)$Individual)
## Dissimilarity: bray
```

```
##
## ANOSIM statistic R: 0.7908
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0803 0.1054 0.1284 0.1517
##
## Dissimilarity ranks between and within classes:
##         0%    25%    50%     75% 100%    N
## Between 44 480.25  909.5 1337.75 1770 1710
## 1       20  21.00   22.0   23.50   25    3
## 2       12  21.50   31.0   34.00   37    3
## 3        3  18.00   33.0   33.50   34    3
## 4        7  31.00   55.0   58.00   61    3
## 5        8  21.50   35.0   35.50   36    3
## 6        2  13.00   24.0   27.00   30    3
## 7       16  66.50  117.0  680.50 1244    3
## 8       10  19.00   28.0   28.50   29    3
## 9       41  42.00   43.0  744.50 1446    3
## 10      17  78.50  140.0  159.50  179    3
## 11       4  13.50   23.0   24.50   26    3
## 12      38  46.00   54.0  272.00  490    3
## 13      15  31.00   47.0  361.50  676    3
## 14      21  30.00   39.0   40.50   42    3
## 15      27 724.50 1422.0 1578.00 1734    3
## 16       1   7.00   13.0   13.50   14    3
## 17       6  12.00   18.0   25.00   32    3
## 18      40 860.00 1680.0 1718.50 1757    3
## 19       5   7.00    9.0   10.00   11    3
## 20      19 120.00  221.0  256.00  291    3
```

```
plot(BCanoIndNR)
```

```
## Warning in bxp(list(stats = structure(c(44, 480, 909.5, 1338, 1770, 20, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```




```
BCps.disperNR <- betadisper(bray_distNR, sample_data(ps)$Sample_type)
anova(BCps.disperNR)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq   Mean Sq F value  Pr(>F)
## Groups     2 0.00940 0.0046998  2.4276 0.09733 .
## Residuals 57 0.11035 0.0019360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(BCps.disperNR)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.00940 0.0046998 2.4276    999  0.107
## Residuals 57 0.11035 0.0019360
```

```
permutest(BCps.disperNR, pairwise = TRUE)
```

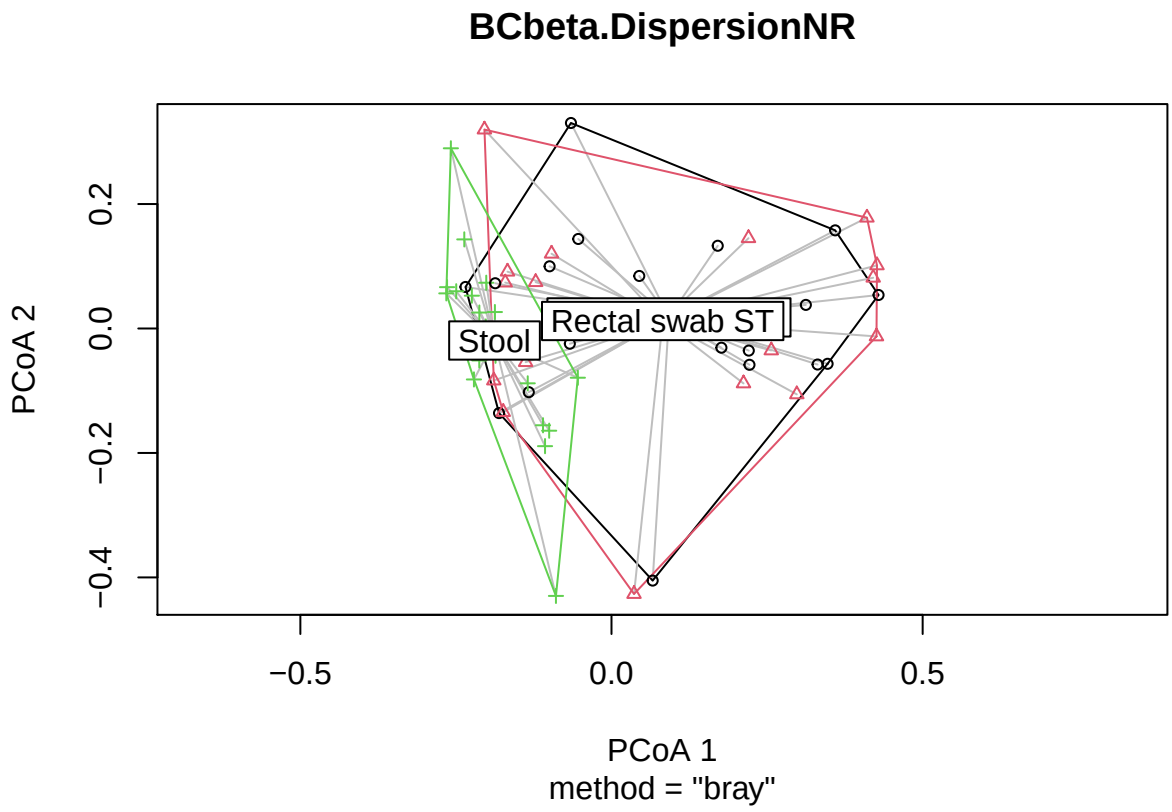
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq   Mean Sq      F N.Perm Pr(>F)
```

```
## Groups     2 0.00940 0.0046998 2.4276    999  0.125
## Residuals 57 0.11035 0.0019360
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                Rectal swab CT Rectal swab ST Stool
## Rectal swab CT                       0.254000 0.328
## Rectal swab ST       0.258058                 0.037
## Stool                0.293984       0.034827
```

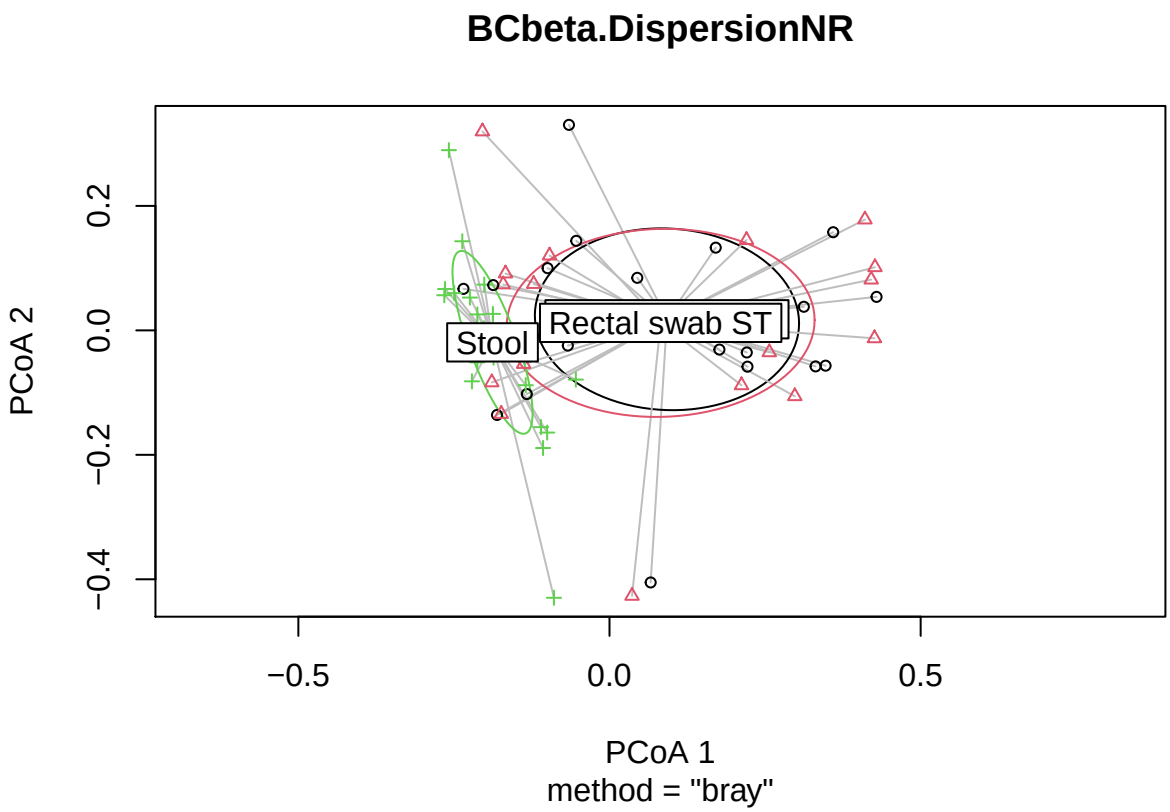
```
TukeyHSD(BCps.disperNR)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                       diff         lwr         upr     p adj
## Rectal swab ST-Rectal swab CT  0.01565203 -0.01783114 0.049135193 0.5028599
## Stool-Rectal swab CT          -0.01500445 -0.04848762 0.018478714 0.5312725
## Stool-Rectal swab ST          -0.03065648 -0.06413964 0.002826686 0.0792059
```

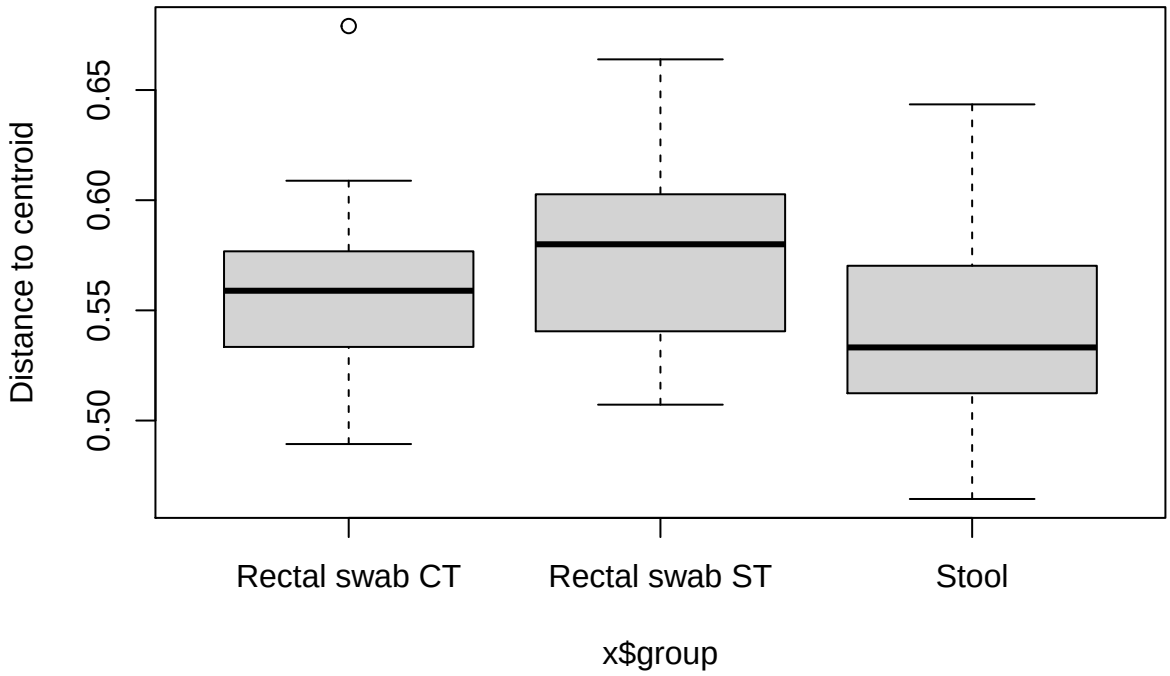
```
# Beta Dispersion Plots
BCbeta.DispersionNR <- BCps.disperNR
plot(BCbeta.DispersionNR)
```



```
plot(BCbeta.DispersionNR, hull = FALSE, ellipse = TRUE)
```

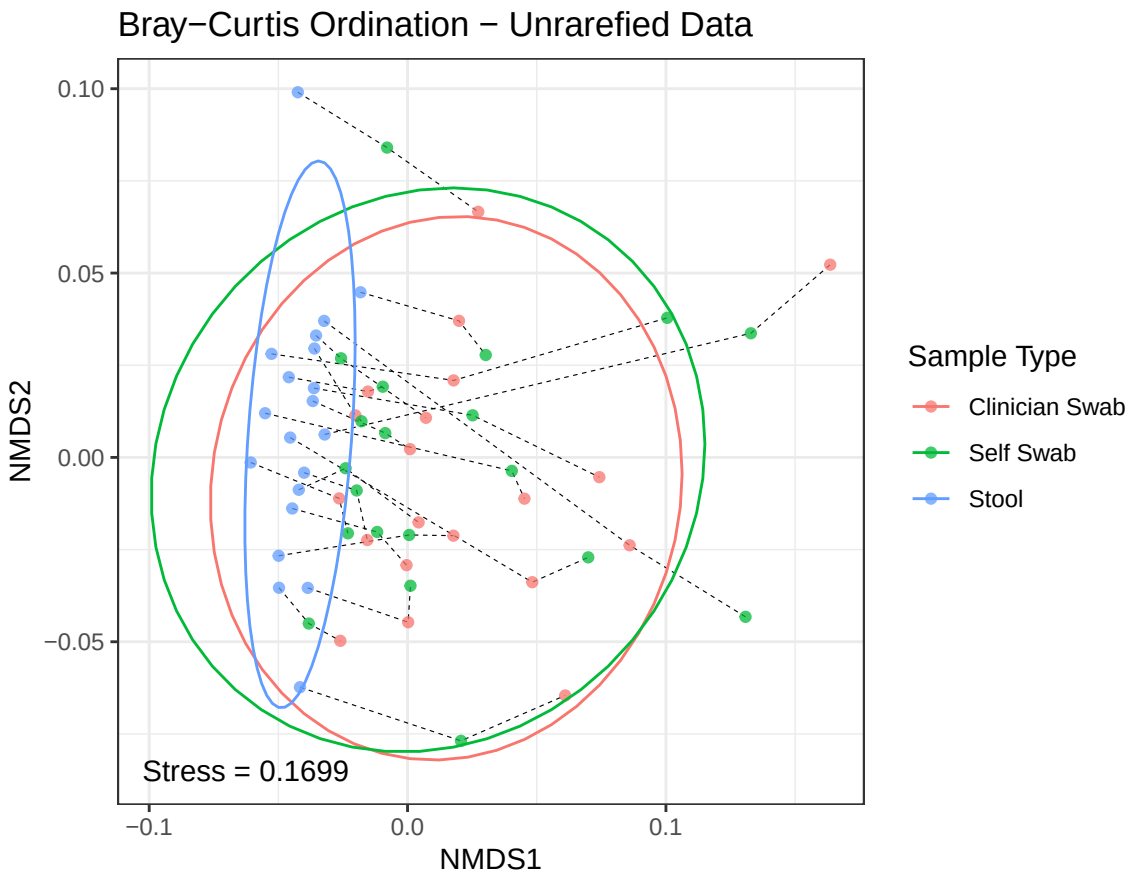



```
boxplot(BCbeta.DispersionNR)
```




```
# Bray-Curtis NMDS plot
bcdfNR <- plot_ordination(ps, ord.nmds.brayNR, justDF = TRUE)
bcdfNR$Sample_type <- gsub("Rectal swab CT", "Clinician Swab", bcdfNR$Sample_type)
```

```
bcdfNR$Sample_type <- gsub("Rectal swab ST", "Self Swab", bcdfNR$Sample_type)

BC_plotNR <- ggplot(bcdfNR, aes(x = NMDS1, y = NMDS2)) +
  geom_line(aes(group = Individual), size = 0.2, linetype = "dashed") +
  geom_point(aes(color = Sample_type), alpha = 0.75) +
  stat_ellipse(aes(color = Sample_type)) +
  annotate("text", x = -0.08, y = -0.085, label = "Stress = ") +
  annotate("text", x = -0.04, y = -0.085, label = round(ord.nmds.brayNR$stress, 4)) +
  labs(title = "Bray-Curtis Ordination - Unrarefied Data", color = "Sample Type") +
  theme(aspect.ratio = 1, plot.margin = unit(c(0, 1, 0, 0), "pt"))

BC_plotNR
```




```
ggsave("../Results/2A)Beta_Diversity_BC_NR.pdf", width = 6, height = 4.5)
```

Beta Diversity - Weighted UniFrac

```
# Ordinate data using Non-metric multidimensional scaling (NMDS) on Weighted Unifrac dissimilarity (dis
uni_dist <- phyloseq::distance(ps_rare, method = "wunifrac")
```

```
## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root as --
## ASV3348 -- in the phylogenetic tree in the data you provided.
```

```
ord.nmds.uni <- ordinate(ps_rare, "NMDS", "wunifrac")
```

```
## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root as --
```

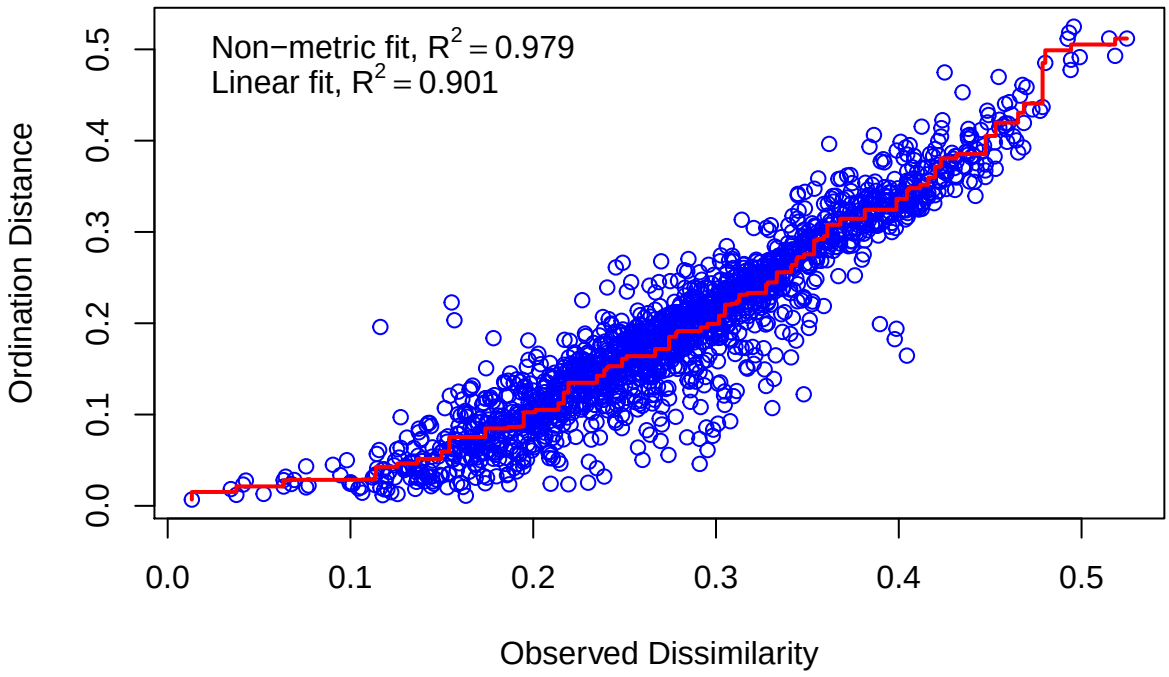
```
## ASV733 -- in the phylogenetic tree in the data you provided.
```

```
## Run 0 stress 0.1432104
## Run 1 stress 0.1474999
## Run 2 stress 0.1800663
## Run 3 stress 0.145561
## Run 4 stress 0.1997951
## Run 5 stress 0.2116014
## Run 6 stress 0.1504798
## Run 7 stress 0.1453868
## Run 8 stress 0.145561
## Run 9 stress 0.143447
## ... Procrustes: rmse 0.006840929  max resid 0.04920322
## Run 10 stress 0.1467449
## Run 11 stress 0.1512407
## Run 12 stress 0.1453868
## Run 13 stress 0.1845533
## Run 14 stress 0.145561
## Run 15 stress 0.2003517
## Run 16 stress 0.1994239
## Run 17 stress 0.1474999
## Run 18 stress 0.1532851
## Run 19 stress 0.1453784
## Run 20 stress 0.1432104
## ... New best solution
## ... Procrustes: rmse 6.537795e-06  max resid 4.583576e-05
## ... Similar to previous best
## *** Solution reached
```

```
# Call newly created file to get the stress value of the plot
ord.nmds.uni
```

```
##
## Call:
## metaMDS(comm = ps.dist)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     ps.dist
## Distance: user supplied
##
## Dimensions: 2
## Stress:     0.1432104
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation
## Species: scores missing
```

```
# Stress plot
stressplot(ord.nmds.uni)
```




```
# Stats
# Test whether the sample types differ significantly from each other using PERMANOVA
adonis(uni_dist ~ sample_data(ps_rare)$Sample_type)
```

```
##
## Call:
## adonis(formula = uni_dist ~ sample_data(ps_rare)$Sample_type)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                 Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
## sample_data(ps_rare)$Sample_type  2   0.49943 0.24971  5.3841 0.1589  0.001 ***
## Residuals                        57   2.64366 0.04638         0.8411
## Total                            59   3.14309                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(formula = uni_dist ~ sample_data(ps_rare)$Individual)
```

```
##
## Call:
## adonis(formula = uni_dist ~ sample_data(ps_rare)$Individual)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## sample_data(ps_rare)$Individual 19    1.9719 0.10378  3.5445 0.62737  0.001 ***
```

```
## Residuals                       40    1.1712 0.02928         0.37263
## Total                           59    3.1431                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(uni_dist ~ sample_data(ps_rare)$Sample_type*sample_data(ps_rare)$Individual)
```

```
##
## Call:
## adonis(formula = uni_dist ~ sample_data(ps_rare)$Sample_type *      sample_data(ps_rare)$Individual)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                                                                 Df SumsOfSqs
## sample_data(ps_rare)$Sample_type                                 2   0.49943
## sample_data(ps_rare)$Individual                                 19   1.97188
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 38   0.67178
## Residuals                                                        0   0.00000
## Total                                                           59   3.14309
##                                                                 MeanSqs
## sample_data(ps_rare)$Sample_type                                      0
## sample_data(ps_rare)$Individual                                       0
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual      0
## Residuals                                                          -Inf
## Total
##                                                                 F.Model
## sample_data(ps_rare)$Sample_type                                      0
## sample_data(ps_rare)$Individual                                       0
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual      0
## Residuals
## Total
##                                                                      R2 Pr(>F)
## sample_data(ps_rare)$Sample_type                                0.15890      1
## sample_data(ps_rare)$Individual                                 0.62737      1
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 0.21373      1
## Residuals                                                       0.00000
## Total                                                           1.00000
```

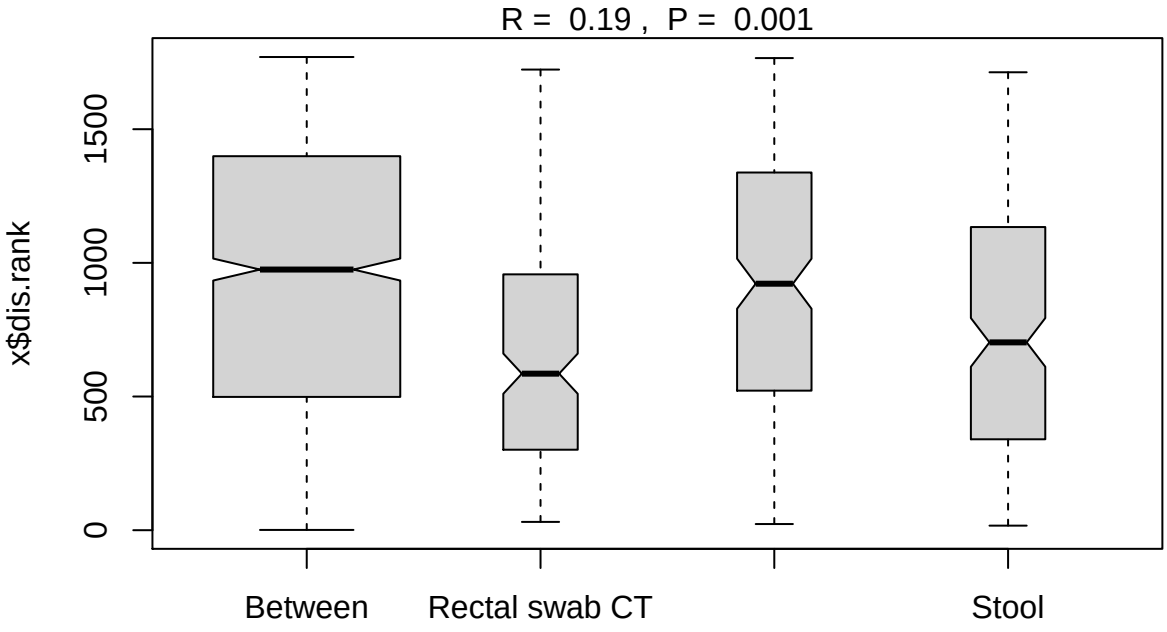
```
anosim(uni_dist, sample_data(ps_rare)$Sample_type)
```

```
##
## Call:
## anosim(x = uni_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity:
##
## ANOSIM statistic R: 0.1896
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

```
UWFanoSamp <- (anosim(uni_dist, sample_data(ps_rare)$Sample_type))
summary(UWFanoSamp)
```

```
##
## Call:
## anosim(x = uni_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity:
##
## ANOSIM statistic R: 0.1896
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0335 0.0462 0.0567 0.0833
##
## Dissimilarity ranks between and within classes:
##                0%    25%   50%    75% 100%    N
## Between         1 498.75 975.0 1398.5 1770 1200
## Rectal swab CT 31 301.50 585.5  955.5 1723  190
## Rectal swab ST 23 525.25 922.0 1337.0 1766  190
## Stool          17 342.25 702.5 1131.5 1713  190
```

```
plot(UWFanoSamp)
```




```
anosim(uni_dist, sample_data(ps_rare)$Individual)
```

```
##
## Call:
## anosim(x = uni_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity:
##
## ANOSIM statistic R: 0.5433
##       Significance: 0.001
```

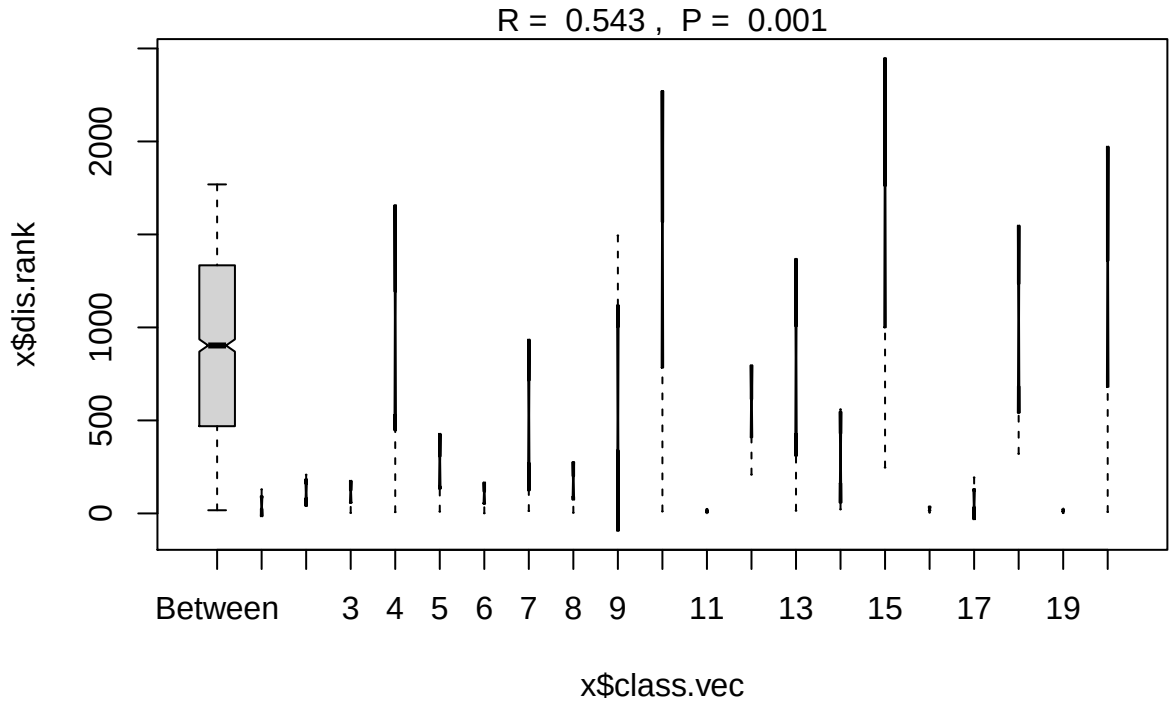
```
##
## Permutation: free
## Number of permutations: 999
```

```
UWFanoInd <- anosim(uni_dist, sample_data(ps_rare)$Individual)
summary(UWFanoInd)
```

```
##
## Call:
## anosim(x = uni_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity:
##
## ANOSIM statistic R: 0.5433
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0792 0.1070 0.1262 0.1559
##
## Dissimilarity ranks between and within classes:
##          0%     25%  50%     75% 100%    N
## Between  17  469.25  903 1333.75 1769 1710
## 1         9   24.00   39   84.00  129    3
## 2        49   80.50  112  160.00  208    3
## 3         3   59.50  116  125.00  134    3
## 4         7  529.50 1052 1193.50 1335    3
## 5        10  145.00  280  307.50  335    3
## 6         1   55.00  109  118.50  128    3
## 7        13  271.00  529  716.00  903    3
## 8         4   89.50  175  201.00  227    3
## 9       163  338.00  513 1003.50 1494    3
## 10       11  784.00 1557 1567.50 1578    3
## 11        6    9.00   12   18.00   24    3
## 12      209  409.50  610  614.50  619    3
## 13       15  427.00  839 1008.00 1177    3
## 14       22  162.00  302  430.50  559    3
## 15      247 1000.50 1754 1762.00 1770    3
## 16        5   15.50   26   28.00   30    3
## 17       14   32.00   50  121.50  193    3
## 18      321  682.50 1044 1235.00 1426    3
## 19        2    9.00   16   17.00   18    3
## 20        8  680.50 1353 1359.50 1366    3
```

```
plot(UWFanoInd)
```

```
## Warning in bxp(list(stats = structure(c(17, 469, 903, 1334, 1769, 9, 24, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```




```
UWFps.disper <- betadisper(uni_dist, sample_data(ps_rare)$Sample_type)
anova(UWFps.disper)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df   Sum Sq   Mean Sq F value Pr(>F)
## Groups     2 0.009689 0.0048446  1.7125 0.1896
## Residuals 57 0.161253 0.0028290
```

```
permutest(UWFps.disper)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.009689 0.0048446 1.7125    999  0.178
## Residuals 57 0.161253 0.0028290
```

```
permutest(UWFps.disper, pairwise = TRUE)
```

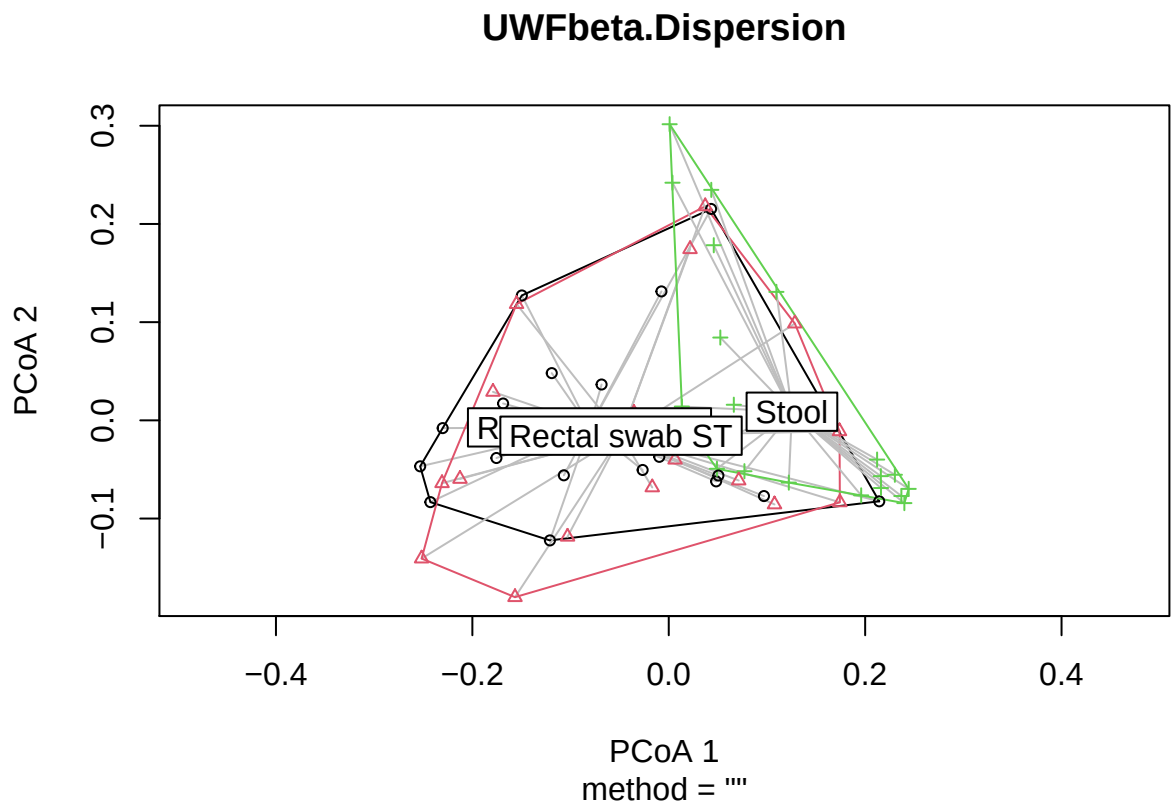
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.009689 0.0048446 1.7125    999   0.18
## Residuals 57 0.161253 0.0028290
```

```
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                 Rectal swab CT Rectal swab ST Stool
## Rectal swab CT                       0.053000 0.492
## Rectal swab ST       0.071517                 0.279
## Stool                0.476931       0.273345
```

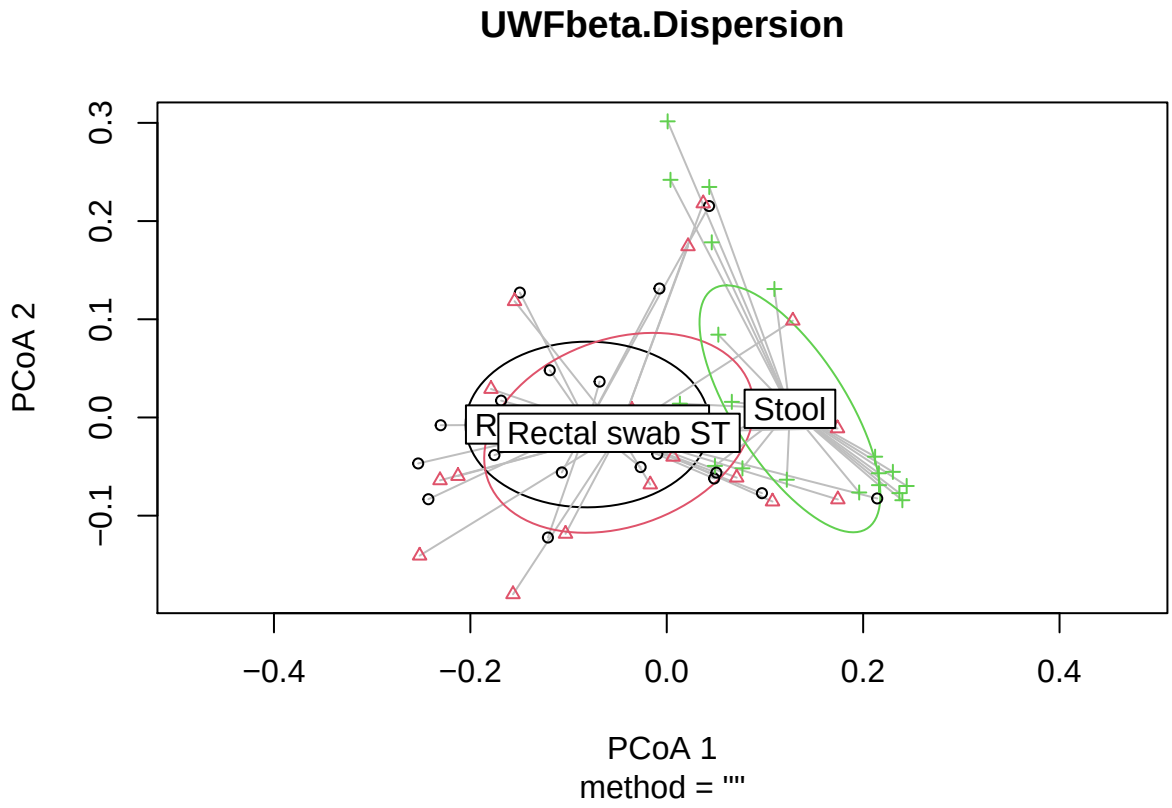
```
TukeyHSD(UWFps.disper)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                     diff          lwr        upr     p adj
## Rectal swab ST-Rectal swab CT  0.03089706 -0.009577969 0.07137209 0.1668541
## Stool-Rectal swab CT           0.01217424 -0.028300789 0.05264927 0.7504612
## Stool-Rectal swab ST          -0.01872282 -0.059197850 0.02175221 0.5099916
```

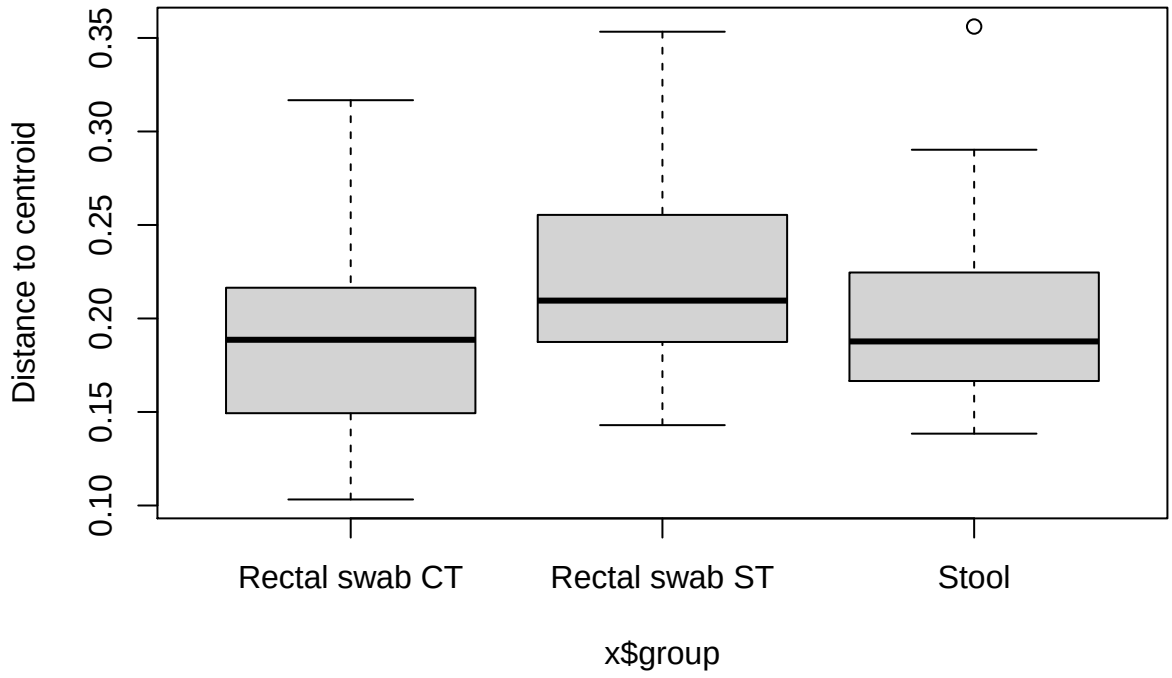
```
# Beta Dispersion Plots
UWFbeta.Dispersion <- UWFps.disper
plot(UWFbeta.Dispersion)
```




```
plot(UWFbeta.Dispersion, hull = FALSE, ellipse = TRUE)
```

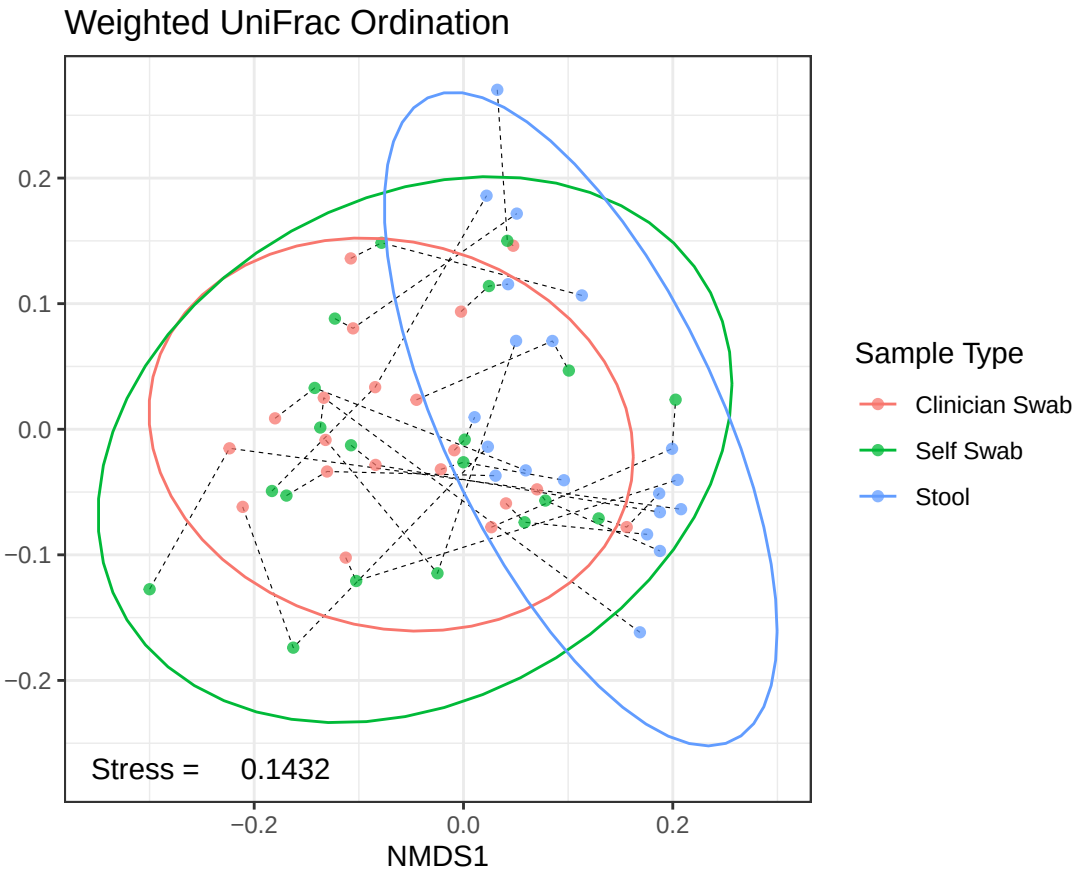



```
boxplot(UWFbeta.Dispersion)
```



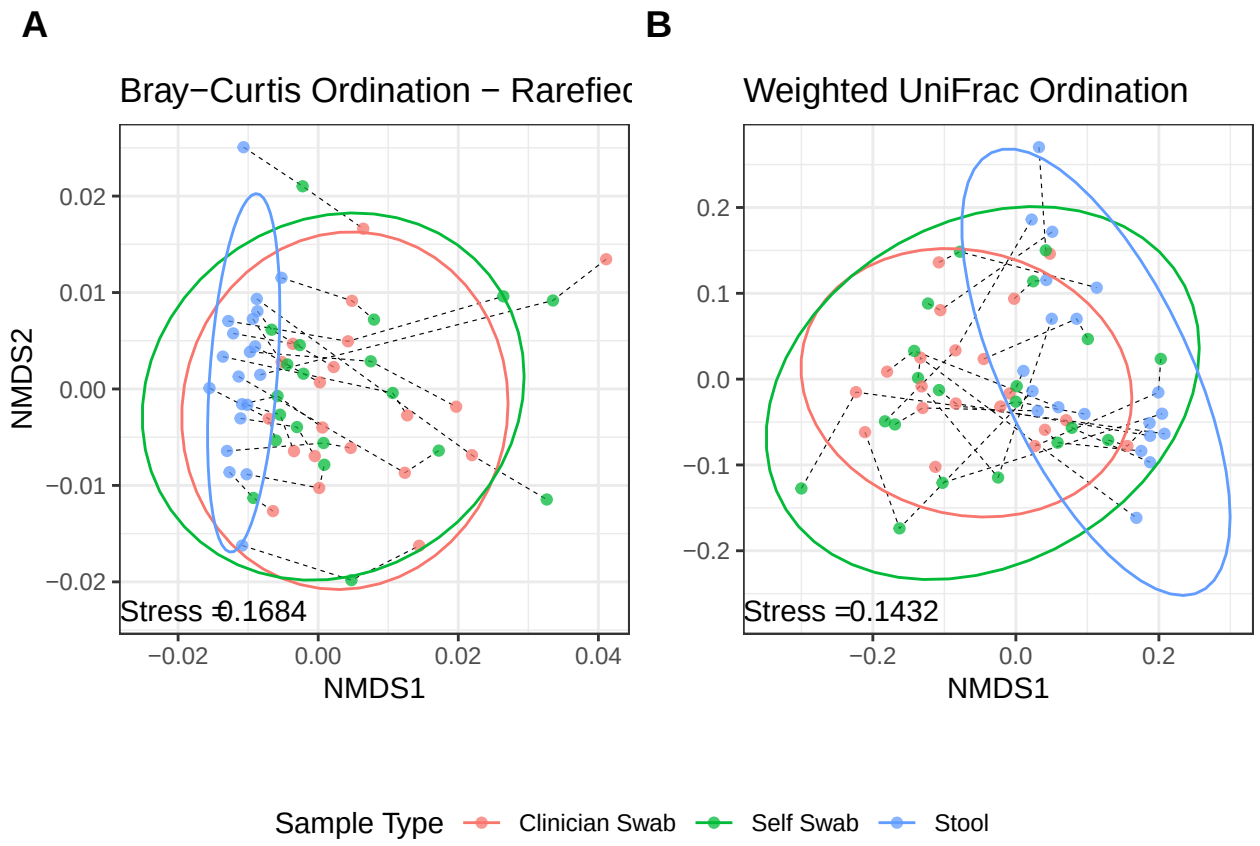

```
# UniFrac NMDS Plot
wuni <- plot_ordination(ps_rare, ord.nmds.uni, justDF = TRUE)
wuni$Sample_type <- gsub("Rectal swab CT", "Clinician Swab", wuni$Sample_type)
wuni$Sample_type <- gsub("Rectal swab ST", "Self Swab", wuni$Sample_type)
```

```
UWF_plot <- ggplot(wuni, aes(x = NMDS1, y = NMDS2)) +
  geom_line(aes(group = Individual), size = 0.2, linetype = "dashed") +
  geom_point(aes(color = Sample_type), alpha = 0.75) +
  stat_ellipse(aes(color = Sample_type)) +
  annotate("text", x = -0.30, y = -0.27, label = "Stress = ") +
  annotate("text", x = -0.17, y = -0.27, label = round(ord.nmds.uni$stress, 4)) +
  labs(title = "Weighted UniFrac Ordination", color = "Sample Type") +
  theme(aspect.ratio = 1, plot.margin = unit(c(0, 0, 0, 1), "pt"),
        axis.title.y = element_blank())

UWF_plot
```




```
ggsave("../Results/2B)Beta_Diversity_wUni.pdf", width = 6, height = 4.5)
```

```
ggarrange(BC_plot, UWF_plot, common.legend = TRUE, legend = c("bottom"),
          align = ("hv"), labels = "AUTO")
```



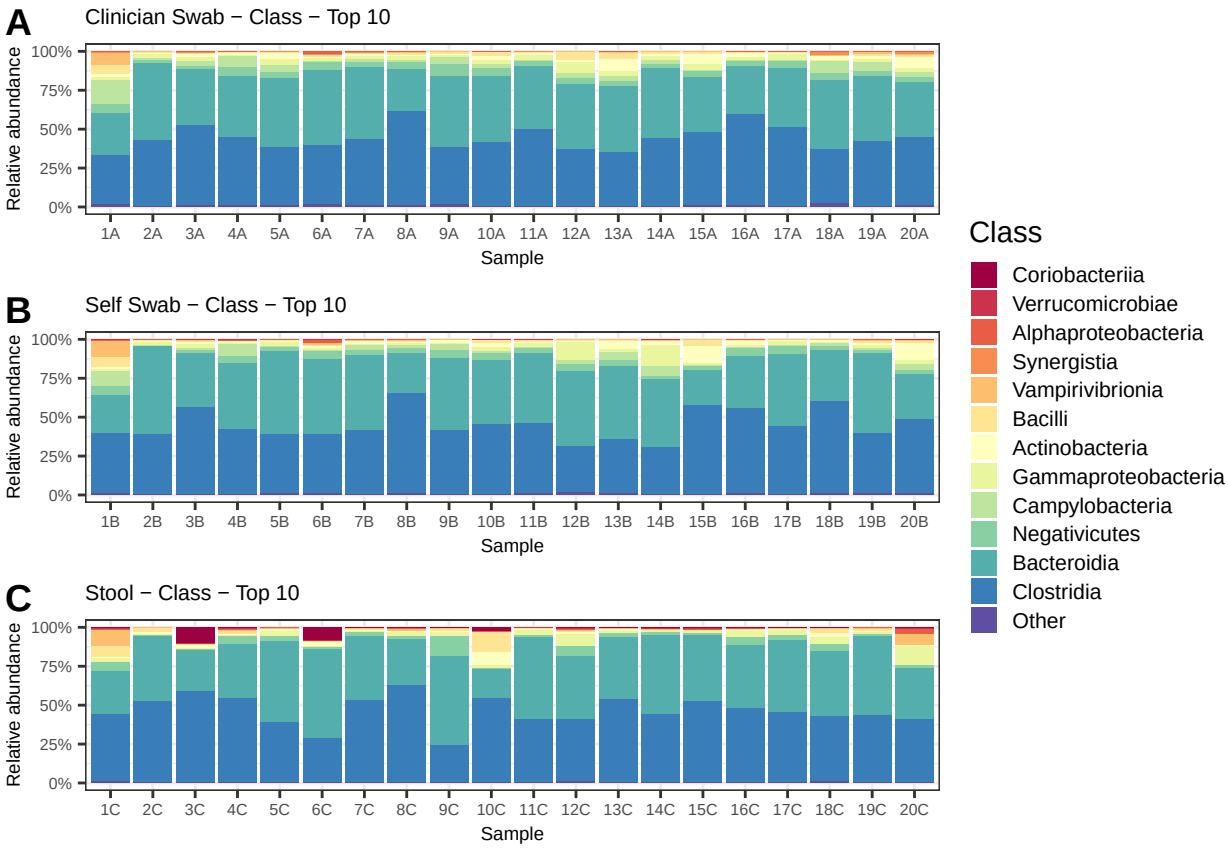

```
ggsave("../Results/2)Beta_Diversity.pdf", width = 8.5, height = 4.5)
```

Relative Abundance - Using Taxonomic Level Class

```
# Subset Phyloseq Objects
ps_class <- subset_taxa(ps_rare, Class != "NA")

sample_clin <- subset_samples(ps_class, Sample_type == "Rectal swab CT")
sample_self <- subset_samples(ps_class, Sample_type == "Rectal swab ST")
sample_stool <- subset_samples(ps_class, Sample_type == "Stool")

# Relative Abundance - Clinician Taken Swab
clin_class <- tax_glom(sample_clin, taxrank = "Class") # agglomerate taxa
clin_class <- transform_sample_counts(clin_class, function(x) x/sum(x)) #get abundance in %
clin_melt <- psmelt(clin_class) # create dataframe from phyloseq object
clin_melt$Class <- as.character(clin_melt$Class) #convert to character
clin_melt <- clin_melt[order(-clin_melt$Abundance),]
clin_melt[!clin_melt$Class %in% c(unique(clin_melt$Class)[1:10]), "Class"] <- "Other"

# Relative Abundance - Self Taken Swab
self_class <- tax_glom(sample_self, taxrank = "Class") # agglomerate taxa
self_class <- transform_sample_counts(self_class, function(x) x/sum(x)) #get abundance in %
self_melt <- psmelt(self_class) # create dataframe from phyloseq object
self_melt$Class <- as.character(self_melt$Class) #convert to character
self_melt <- self_melt[order(-self_melt$Abundance),]
self_melt[!self_melt$Class %in% c(unique(self_melt$Class)[1:10]), "Class"] <- "Other"
```

```
# Relative Abundance - Stool Sample
stool_class <- tax_glom(sample_stool, taxrank = "Class") # agglomerate taxa
stool_class <- transform_sample_counts(stool_class, function(x) x/sum(x)) #get abundance in %
stool_melt <- psmelt(stool_class) # create dataframe from phyloseq object
stool_melt$Class <- as.character(stool_melt$Class) #convert to character
stool_melt <- stool_melt[order(-stool_melt$Abundance),]
stool_melt[!stool_melt$Class %in% c(unique(stool_melt$Class)[1:10]), "Class"] <- "Other"


# Set order of bars and get colours
sort.clin <- clin_melt %>%
  plyr::count("Class", wt = "Abundance") %>%
  arrange(desc(freq)) %>%
  pull(Class)
sort.clin <- sort.clin[!sort.clin %in% "Other"]
sort.clin <- append("Other", sort.clin)

sort.self <- self_melt %>%
  plyr::count("Class", wt = "Abundance") %>%
  arrange(desc(freq)) %>%
  pull(Class)
sort.self <- sort.self[!sort.self %in% "Other"]
sort.self <- append("Other", sort.self)

sort.stool <- stool_melt %>%
  plyr::count("Class", wt = "Abundance") %>%
  arrange(desc(freq)) %>%
  pull(Class)
sort.stool <- sort.stool[!sort.stool %in% "Other"]
sort.stool <- append("Other", sort.stool)

barOrder <- unique(c(sort.clin, sort.self, sort.stool))

# Get Colours and Assign to Bacteria
spectralExtra <- colorRampPalette(brewer.pal(11, "Spectral"))(length(barOrder))
cols <- setNames(c(spectralExtra), c(rev(barOrder)))

# Create Custom Legend
dummy_df <- data.frame(
  Class = as.factor(barOrder) ,
  value = c(1,2,3,4,5,6,7,8,9,10,11,12,13))
dummy_df <- mutate(dummy_df, Class = factor(Class, levels = rev(barOrder)))

rel_legend <- get_legend(ggplot(dummy_df, aes(x = Class, y = value)) +
    geom_bar(stat = "identity", aes(fill = Class)) +
    scale_fill_manual(values = cols) +
    theme(legend.text = element_text(size = 8), legend.key.size = unit(0.75, "line")))


# Plot - Relative Abundance - Clinician Taken Swab
t1_class <- clin_melt %>%
  mutate(Sample = factor(Sample, levels = c("1A", "2A", "3A", "4A", "5A",
                                            "6A", "7A", "8A", '9A', "10A",
```

```
                                          "11A", "12A", "13A", "14A", "15A",
                                          "16A", "17A", "18A", "19A", "20A"))) %>%
  mutate(Class = factor(Class, levels = rev(barOrder))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme(text = element_text(size = 7)) +
  ggtitle("Clinician Swab - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_manual(values = cols) + theme(legend.position = "none")

# Plot - Relative Abundance - Self Taken Swab
t2_class <- self_melt %>%
  mutate(Sample = factor(Sample, levels = c("1B", "2B", "3B", "4B", "5B",
                                          "6B", "7B", "8B", "9B", "10B",
                                          "11B", "12B", "13B", "14B", "15B",
                                          "16B", "17B", "18B", "19B", "20B"))) %>%
  mutate(Class = factor(Class, levels = rev(barOrder))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme(text = element_text(size = 7)) +
  ggtitle("Self Swab - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_manual(values = cols) + theme(legend.position = "none")

# Plot - Relative Abundance - Stool Sample
t3_class <- stool_melt %>%
  mutate(Sample = factor(Sample, levels = c("1C", "2C", "3C", "4C", "5C",
                                          "6C", "7C", "8C", "9C", "10C",
                                          "11C", "12C", "13C", "14C", "15C",
                                          "16C", "17C", "18C", "19C", "20C"))) %>%
  mutate(Class = factor(Class, levels = rev(barOrder))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme(text = element_text(size = 7)) +
  ggtitle("Stool - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_manual(values = cols) + theme(legend.position = "none")

plots <- ggarrange(t1_class, t2_class, t3_class, nrow = 3, labels = "AUTO")
ggarrange(plots, legend.grob = rel_legend, legend = "right")
```




```
ggsave("../Results/3)Relative_Abundance.pdf", width = 7, height = 8)
```

OTU differential abundance testing with DESeq2

```
ps_deseq <- ps %>%
  tax_glom(taxrank = "Genus")

sample_data(ps_deseq)$Sample_type <- gsub(" ", "_", sample_data(ps_deseq)$Sample_type)
sample_data(ps_deseq)$Sample_type <- as.factor(sample_data(ps_deseq)$Sample_type)

# Convert the phyloseq object to a DESeqDataSet
ds <- phyloseq_to_deseq2(ps_deseq, ~ Sample_type)
```

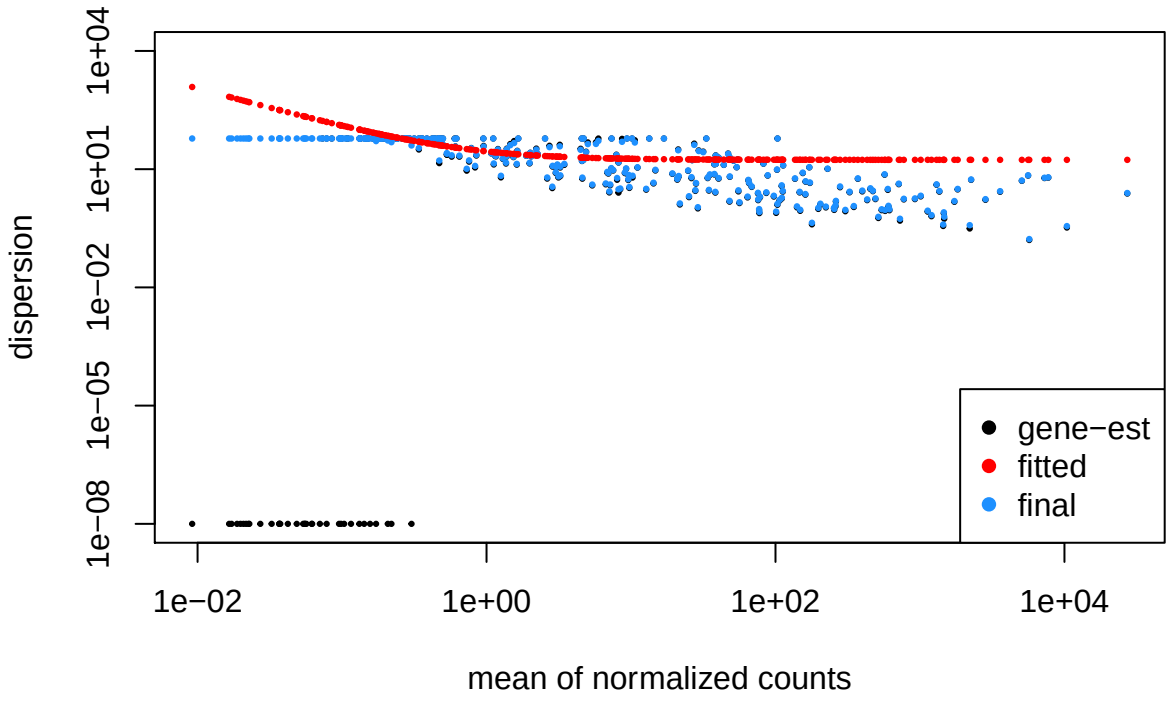
```
## converting counts to integer mode
```

```
ds <- DESeq(ds)
```

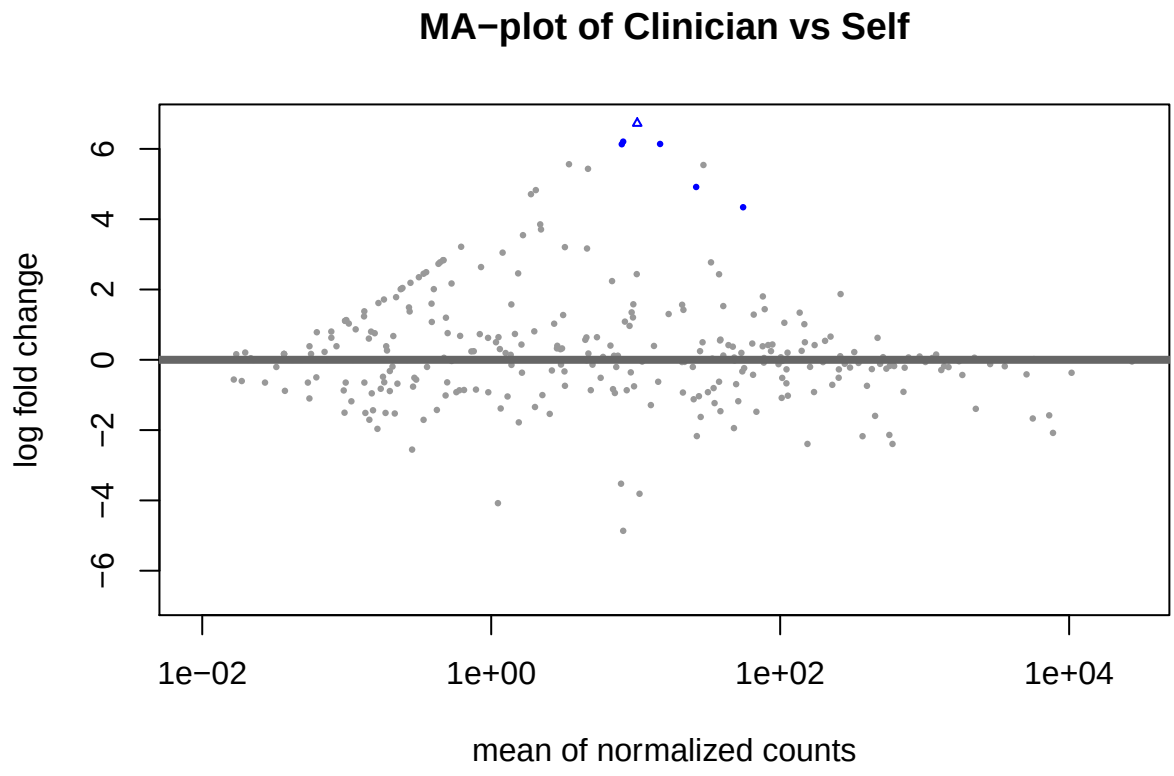
```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## -- replacing outliers and refitting for 151 genes
```

```
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

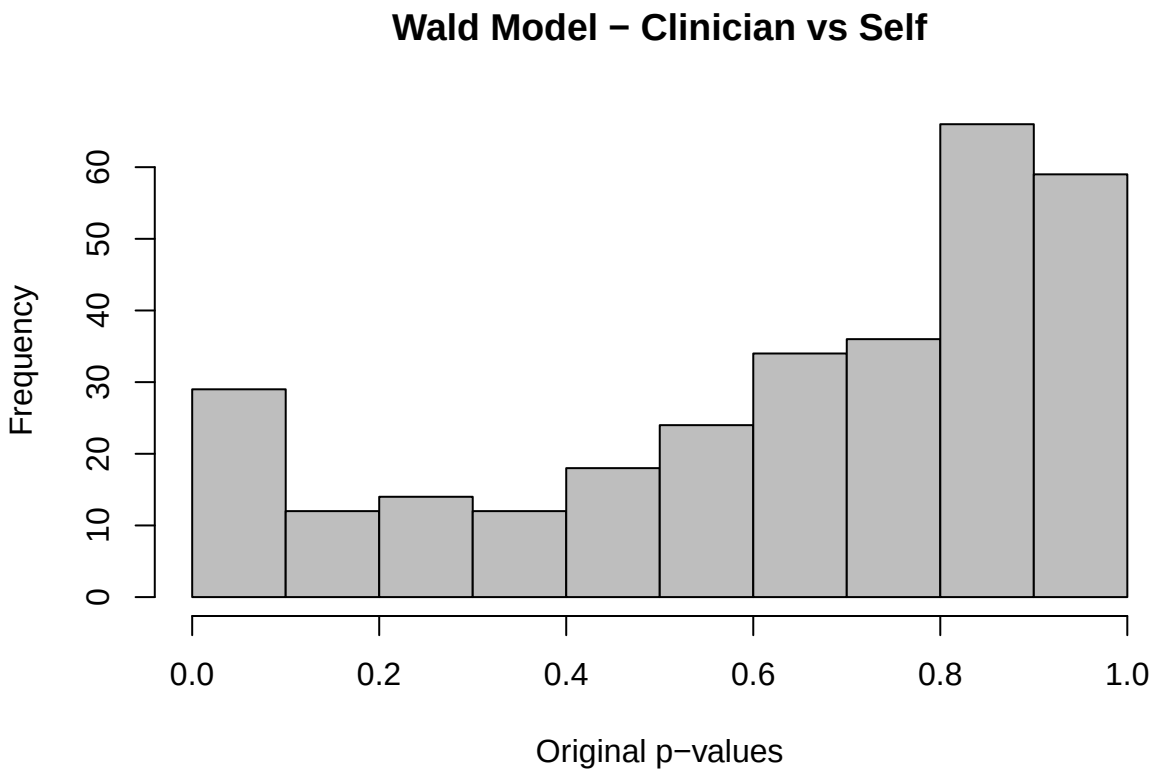
```
# Plot of Dispersion Estimates
plotDispEsts(ds, ylim = c(1e-8, 1e4))
```




```
# Extract the result table from the ds object usind the DESeq2 function results and filter the OTUs usi
alpha <- 0.01

# Swab CT vs Swab ST
resCTST <- results(ds, contrast = c("Sample_type", "Rectal_swab_CT", "Rectal_swab_ST"),
                   alpha = alpha)
resCTST <- resCTST[order(resCTST$padj, na.last = NA), ]
plotMA(resCTST, alpha = 0.01, main = "MA-plot of Clinician vs Self")
```



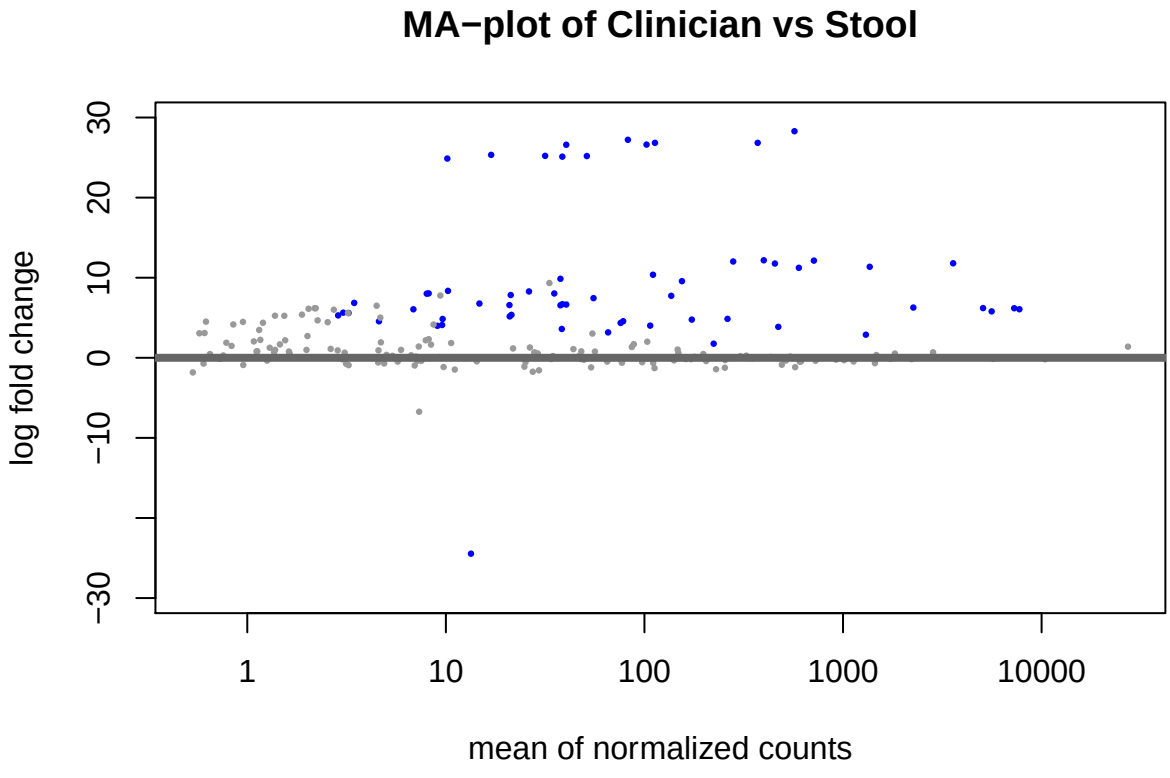

```
hist(resCTST$pvalue, col = "gray", main = "Wald Model - Clinician vs Self", xlab = "Original p-values")
```



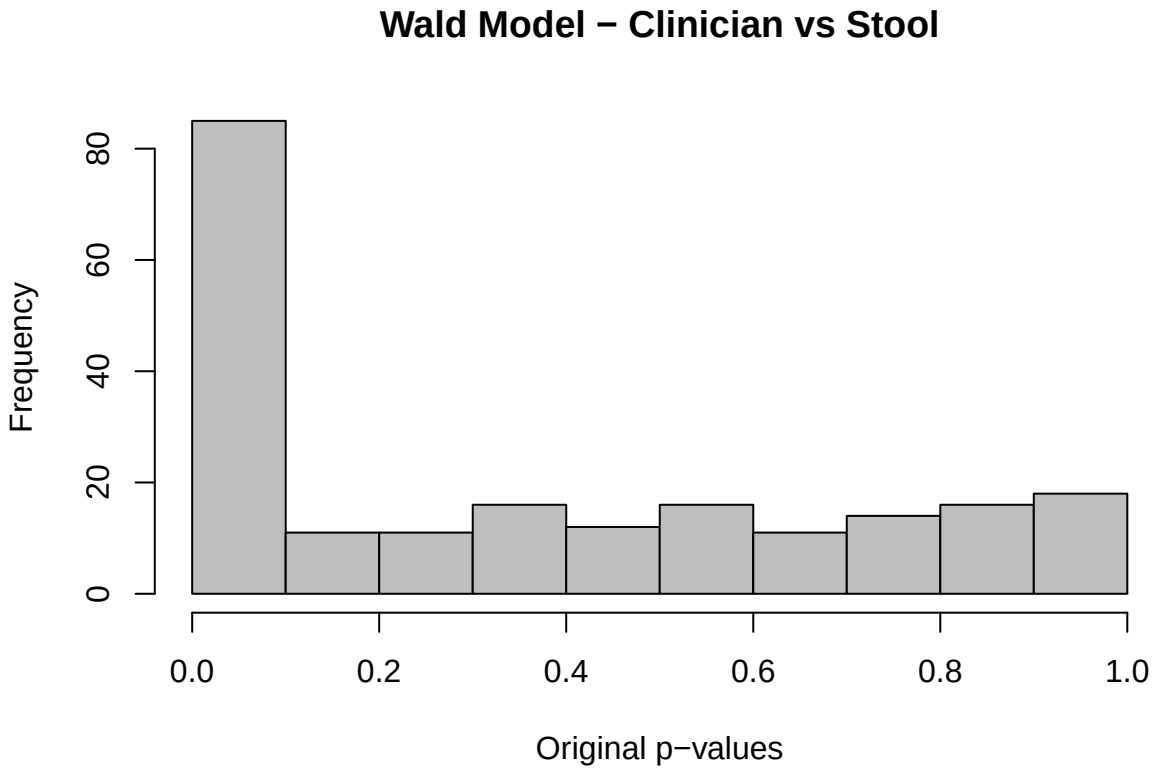

```
resCTST_sig <- resCTST[(resCTST$padj < alpha), ]
resCTST_sig <- cbind(as(resCTST_sig, "data.frame"), as(tax_table(ps)[rownames(resCTST_sig), ], "matrix")
head(resCTST_sig)
```

```
##          baseMean log2FoldChange     lfcSE     stat       pvalue         padj
## ASV1580  8.191810       6.205783 0.9102231 6.817871 9.239974e-12 2.808952e-09
## ASV3522 10.246362       7.072852 1.2769844 5.538714 3.046999e-08 4.631439e-06
## ASV1634 55.398735       4.339785 0.7977984 5.439702 5.336990e-08 5.408149e-06
## ASV3554  8.010744       6.131689 1.1953911 5.129441 2.906035e-07 2.208587e-05
## ASV1485 26.218338       4.915912 1.1726628 4.192093 2.763921e-05 1.473097e-03
## ASV1582 14.753210       6.139537 1.4685785 4.180598 2.907429e-05 1.473097e-03
##          Kingdom         Phylum               Class            Order
## ASV1580 Bacteria Proteobacteria Gammaproteobacteria Enterobacterales
## ASV3522 Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV1634 Bacteria Proteobacteria Gammaproteobacteria  Pseudomonadales
## ASV3554 Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV1485 Bacteria Proteobacteria Gammaproteobacteria    Aeromonadales
## ASV1582 Bacteria Proteobacteria Gammaproteobacteria Enterobacterales
##                  Family                         Genus
## ASV1580     Yersiniaceae                      Yersinia
## ASV3522   Clostridiaceae  Clostridium_sensu_stricto_5
## ASV1634 Pseudomonadaceae                   Pseudomonas
## ASV3554   Clostridiaceae Clostridium_sensu_stricto_13
## ASV1485   Aeromonadaceae                    Aeromonas
## ASV1582       Hafniaceae       Hafnia-Obesumbacterium
```

```r
# Swab CT vs Stool
resCTS <- results(ds, contrast = c("Sample_type", "Rectal_swab_CT", "Stool"),
                  alpha = alpha)
resCTS <- resCTS[order(resCTS$padj, na.last = NA), ]
plotMA(resCTS, alpha = 0.01, main = "MA-plot of Clinician vs Stool")
```



```
hist(resCTS$pvalue, col = "gray", main = "Wald Model - Clinician vs Stool", xlab = "Original p-values")
```

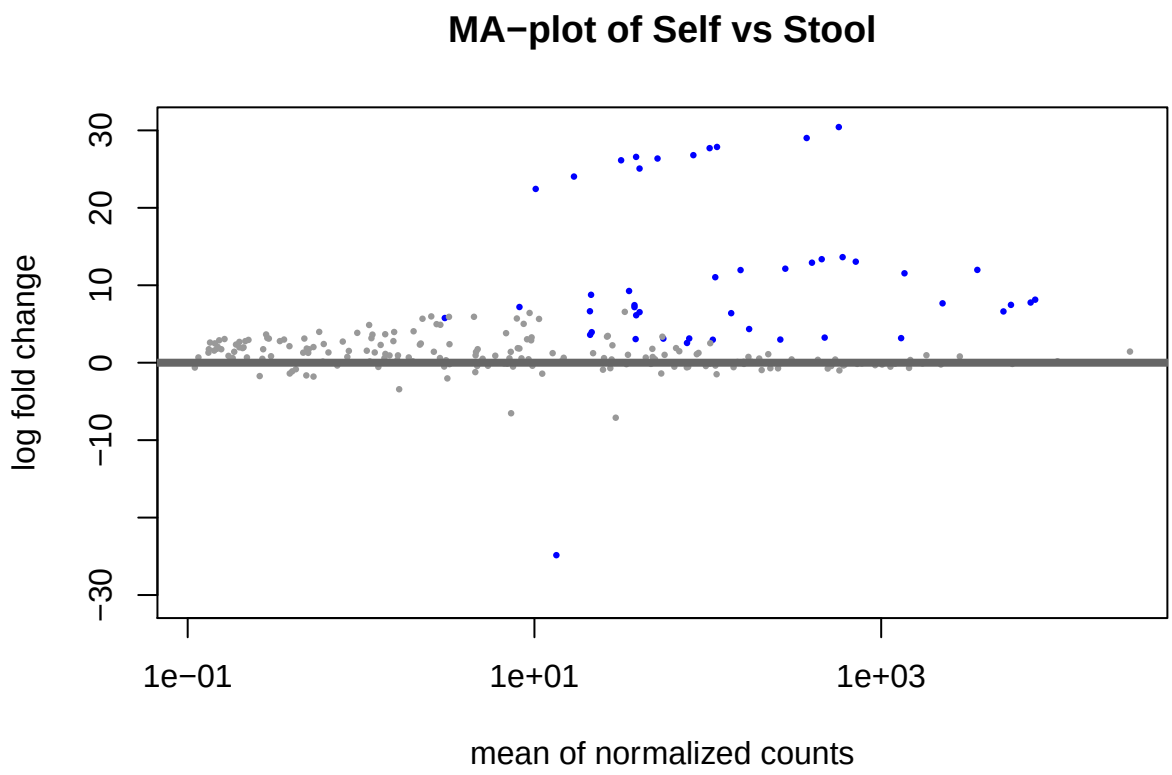



```
resCTS_sig <- resCTS[(resCTS$padj < alpha), ]
resCTS_sig <- cbind(as(resCTS_sig, "data.frame"), as(tax_table(ps)[rownames(resCTS_sig), ], "matrix"))
head(resCTS_sig)
```

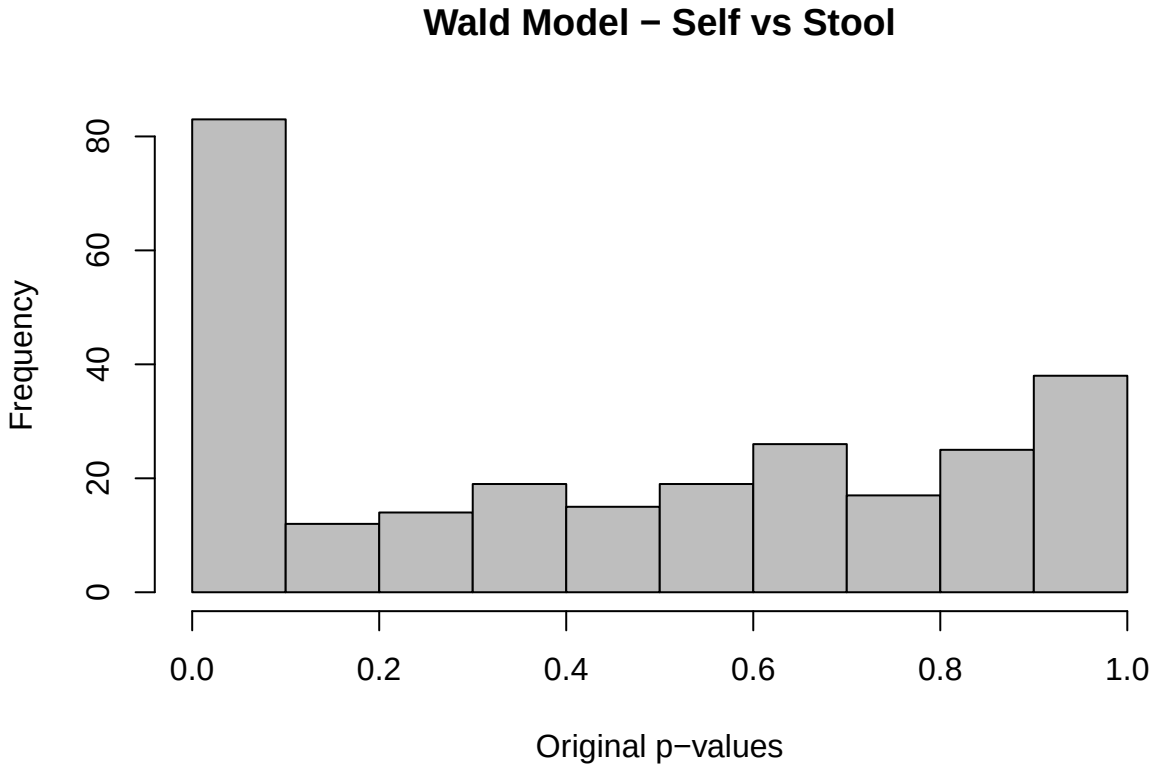
```
##           baseMean log2FoldChange     lfcSE     stat        pvalue
## ASV3460  371.68353       26.83913 1.1950890 22.45785 1.072464e-111
## ASV3599  569.52608       28.29018 1.3237180 21.37176 2.447353e-101
## ASV940   102.45995       26.62138 1.5030199 17.71193  3.392565e-70
## ASV3146  112.98959       26.83745 1.6813612 15.96174  2.360599e-57
## ASV1183   82.54958       27.21862 1.7630615 15.43827  9.049272e-54
## ASV1254 3585.31720       11.79910 0.7864814 15.00239  7.082408e-51
##                  padj  Kingdom           Phylum           Class
## ASV3460 2.252174e-109 Bacteria       Firmicutes   Negativicutes
## ASV3599  2.569720e-99 Bacteria       Firmicutes      Clostridia
## ASV940   2.374796e-68 Bacteria       Firmicutes      Clostridia
## ASV3146  1.239315e-55 Bacteria       Firmicutes         Bacilli
## ASV1183  3.800694e-52 Bacteria     Synergistota     Synergistia
## ASV1254  2.478843e-49 Bacteria Campilobacterota Campylobacteria
##                                    Order
## ASV3460      Veillonellales-Selenomonadales
## ASV3599                        Clostridia_or
## ASV940  Peptostreptococcales-Tissierellales
## ASV3146                     Lactobacillales
## ASV1183                       Synergistales
## ASV1254                   Campylobacterales
##                                     Family          Genus
## ASV3460                     Veillonellaceae Negativicoccus
```

```
## ASV3599                 Hungateiclostridiaceae Fastidiosipila
## ASV940  Peptostreptococcales-Tissierellales_fa      Gallicola
## ASV3146                          Aerococcaceae      Facklamia
## ASV1183                        Synergistaceae Pyramidobacter
## ASV1254                     Campylobacteraceae  Campylobacter
```

```
# Swab ST vs Stool
resSTS <- results(ds, contrast = c("Sample_type", "Rectal_swab_ST", "Stool"),
                  alpha = alpha)
resSTS <- resSTS[order(resSTS$padj, na.last = NA), ]
plotMA(resSTS, alpha = 0.01, main = "MA-plot of Self vs Stool")
```




```
hist(resSTS$pvalue, col = "gray", main = "Wald Model - Self vs Stool", xlab = "Original p-values")
```




```
resSTS_sig <- resSTS[(resSTS$padj < alpha), ]
resSTS_sig <- cbind(as(resSTS_sig, "data.frame"), as(tax_table(ps)[rownames(resSTS_sig), ], "matrix"))
head(resSTS_sig)
```

```
##           baseMean log2FoldChange     lfcSE     stat        pvalue
## ASV3460  371.68353       29.01258 1.1939331 24.30001 1.960067e-130
## ASV3599  569.52608       30.42730 1.3230276 22.99823 4.854894e-117
## ASV940   102.45995       27.70469 1.5015082 18.45124  5.095997e-76
## ASV3146  112.98959       27.85901 1.6801511 16.58125  9.522523e-62
## ASV1254 3585.31720       11.98355 0.7864397 15.23772  1.986922e-52
## ASV1183   82.54958       26.79915 1.7628105 15.20251  3.403086e-52
##                  padj  Kingdom           Phylum           Class
## ASV3460 5.252978e-128 Bacteria       Firmicutes   Negativicutes
## ASV3599 6.505558e-115 Bacteria       Firmicutes      Clostridia
## ASV940   4.552424e-74 Bacteria       Firmicutes      Clostridia
## ASV3146  6.380090e-60 Bacteria       Firmicutes         Bacilli
## ASV1254  1.064990e-50 Bacteria Campilobacterota Campylobacteria
## ASV1183  1.520045e-50 Bacteria     Synergistota     Synergistia
##                                      Order
## ASV3460      Veillonellales-Selenomonadales
## ASV3599                        Clostridia_or
## ASV940  Peptostreptococcales-Tissierellales
## ASV3146                     Lactobacillales
## ASV1254                   Campylobacterales
## ASV1183                       Synergistales
##                                        Family          Genus
## ASV3460                       Veillonellaceae Negativicoccus
## ASV3599                 Hungateiclostridiaceae Fastidiosipila
## ASV940  Peptostreptococcales-Tissierellales_fa      Gallicola
## ASV3146                          Aerococcaceae      Facklamia
```

```
## ASV1254                     Campylobacteraceae  Campylobacter
## ASV1183                         Synergistaceae Pyramidobacter
```

```
# Save .csv of significant fold change results
resCTST_sig$Comparison <- "Clinician Taken Swab vs Self Taken Swab"
resCTS_sig$Comparison <- "Clinician Taken Swab vs Stool"
resSTS_sig$Comparison <- "Self Taken Swab vs Stool"

SignificantResults <- rbind(resCTST_sig, resCTS_sig, resSTS_sig)
write.csv(SignificantResults, file = "../Results/SignificantFoldChangeResults.csv")
```

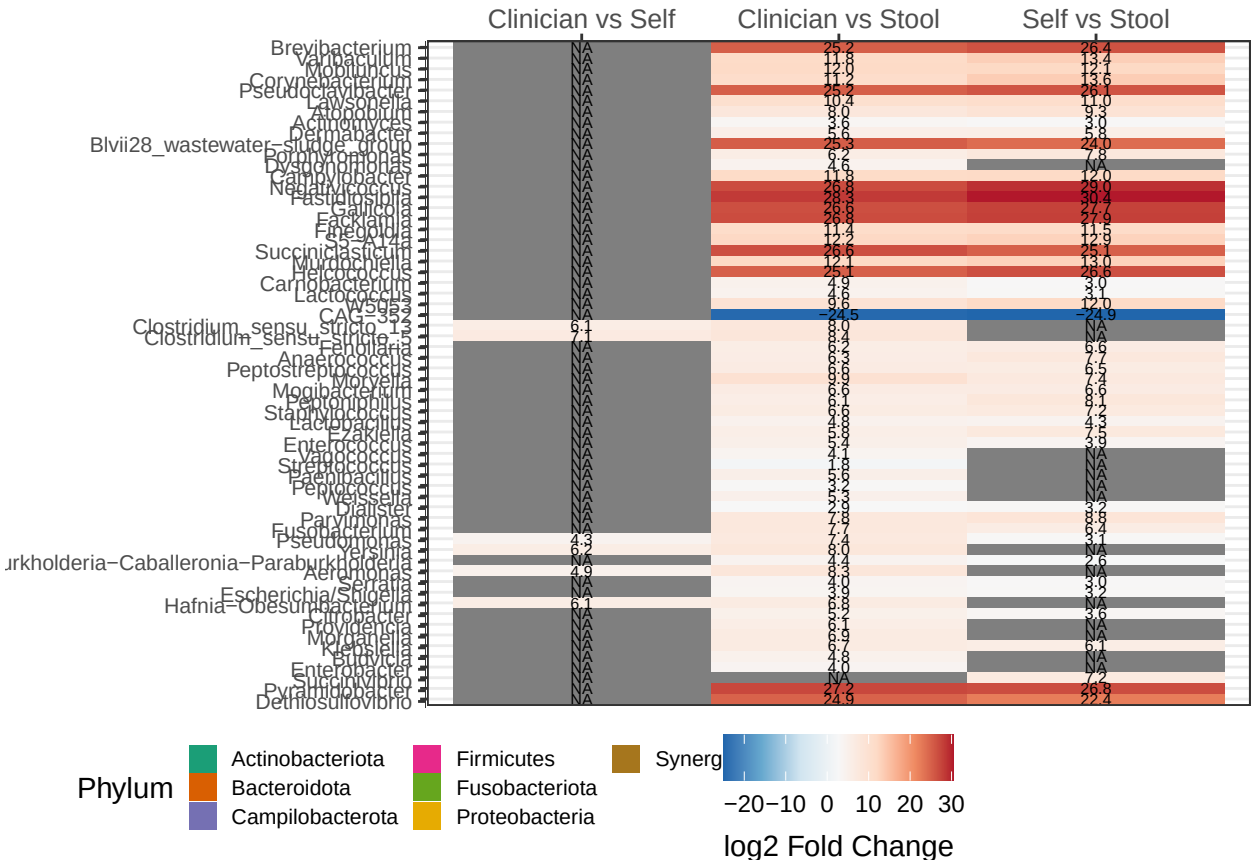
Differential Abundance - ggplot Heatmap

```
diffCTST <- resCTST_sig %>%
  select(log2FoldChange, Phylum, Genus)
colnames(diffCTST)[1] <- "CTST_log2FoldChange"

diffCTS <- resCTS_sig %>%
  select(log2FoldChange, Phylum, Genus)
colnames(diffCTS)[1] <- "CTS_log2FoldChange"

diffSTS <- resSTS_sig %>%
  select(log2FoldChange, Phylum, Genus)
colnames(diffSTS)[1] <- "STS_log2FoldChange"

heat <- rbind.fill(as.data.frame(t(diffCTS)), as.data.frame(t(diffSTS)))
heat <- rbind.fill(as.data.frame(heat), as.data.frame(t(diffCTST)))
heat <- as.data.frame(t(heat))
colnames(heat) <- c("CTS", "CTS_phylum", "CTS_genus",
                    "STS", "STS_phylum", "STS_genus",
                    "CTST", "CTST_phylum", "CTST_genus")

heat$Phylum <- as.character(heat$CTS_phylum)
heat$Phylum[nrow(heat)] <- as.character(heat$STS_phylum[nrow(heat)])

heat$Genus <- as.character(heat$CTS_genus)
heat$Genus[nrow(heat)] <- as.character(heat$STS_genus[nrow(heat)])

# long format needed for ggplot based heat map
heatLong <- heat %>%
  select(Phylum, Genus, CTS, STS, CTST) %>%
  tidyr::pivot_longer(heat, cols = CTS:CTST,
                      names_to = "Contrast", values_to = "log2FC")
```

```
## Warning in gsub(paste0("^", names_prefix), "", names(cols)): argument 'pattern'
## has length > 1 and only the first element will be used
```

```
heatLong$log2FC <- as.numeric(as.character(heatLong$log2FC))
heatLong$Contrast <- factor(heatLong$Contrast, levels = c("CTST", "CTS", "STS"))
heatLong <- heatLong %>%
  arrange(Phylum) %>%
  mutate(Genus = factor(Genus, levels = unique(Genus)))

heatLog <- ggplot(heatLong, aes(Contrast, Genus, fill = log2FC)) + geom_tile() +
```

```
  geom_text(aes(label = sprintf("%2.1f", log2FC)), size = 2) +
  labs(fill = "log2 Fold Change") +
  theme(axis.title = element_blank(), legend.position = "bottom",
        axis.text.x = element_text(family = "Helvetica", size = 10, face = "plain"),
        axis.text.y = element_blank(), plot.background = element_blank(),
        plot.margin = margin(t = 2, r = 0, b = 0, l = 0, unit = "pt"),
        legend.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "pt")) +
  guides(fill = guide_colourbar(title.position = "bottom", title.hjust = 0.5)) +
  scale_fill_distiller(palette = "RdBu") +
  scale_y_discrete(limits = rev) +
  scale_x_discrete(position = "top", labels = (c("Clinician vs Self",
                                                 "Clinician vs Stool",
                                                 "Self vs Stool")))

heatPhylum <- ggplot(heatLong, aes(Contrast, Genus, fill = Phylum)) + geom_tile() +
  scale_y_discrete(limits = rev) +
  theme(axis.title = element_blank(), legend.position = "bottom",
        axis.text.y = element_text(size = 8),
        axis.text.x = element_blank(), axis.ticks.x = element_blank(),
        plot.margin = margin(t = 16.5, r = 5, b = 11, l = 0, unit = "pt"),
        legend.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "pt"),
        legend.text = element_text(size = 8), legend.key.size = unit(0.75, "line")) +
  scale_fill_brewer(palette = "Dark2", guide = guide_legend(ncol = 3))

heatChanges <- ggarrange(heatPhylum, heatLog, widths = c(1, 2))
heatChanges
```



```
ggsave("../Results/4)Differential_Abundance_heatmap.pdf", width = 8, height = 8)
```

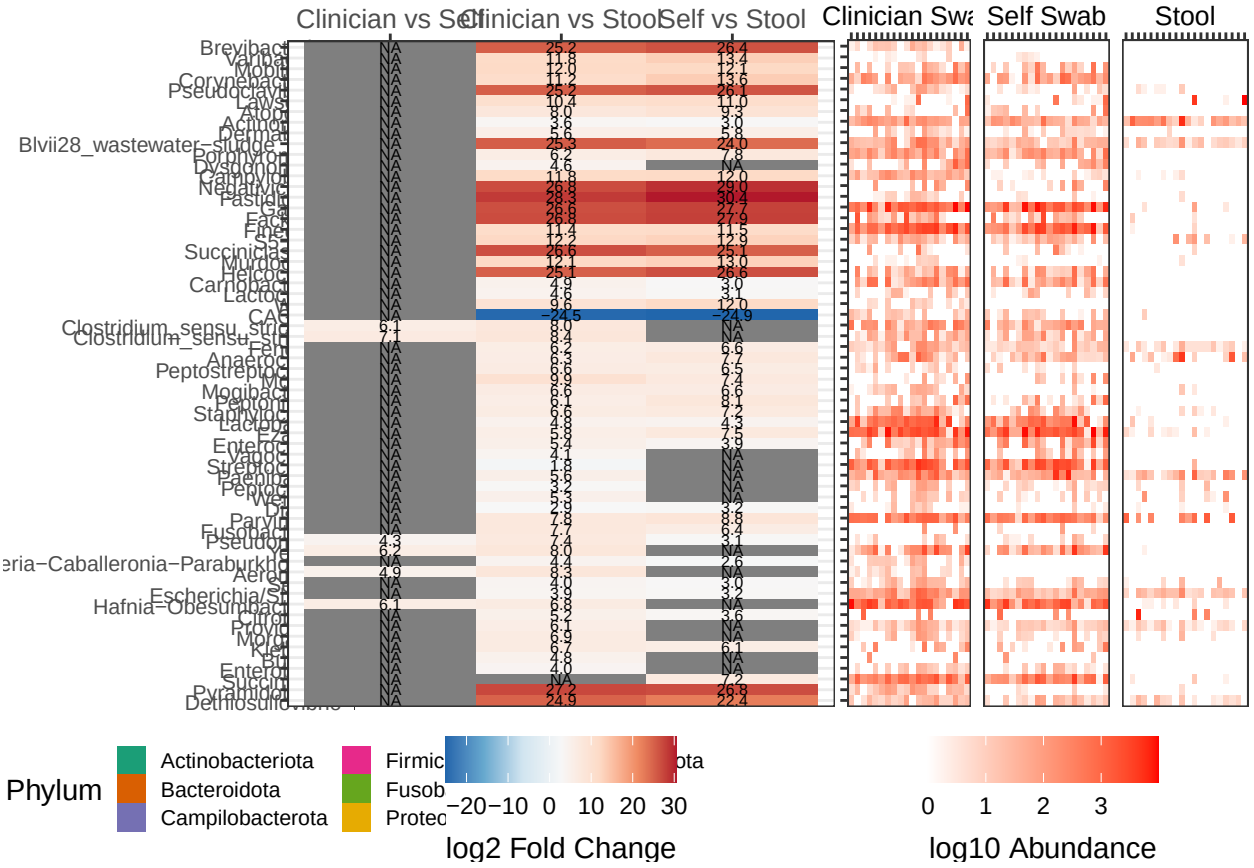
Genus Abundance ggplot Heatmap

```
# Make figure with individual abundance to go next to heat map
heat_ps <- subset_taxa(ps_rare, Genus %in% heat$Genus)
heat_ps <- heat_ps %>%
  tax_glom(taxrank = "Genus")

# Clinician Swab
heat_clin <- subset_samples(heat_ps, Sample_type == "Rectal swab CT")
melted_clin <- psmelt(heat_clin)
melted_clin <- select(melted_clin, Individual, Genus, Abundance)
melted_clin$Abundance[melted_clin$Abundance == 0] <- 1
melted_clin$log2Abundance <- log2(melted_clin$Abundance)
melted_clin$log10Abundance <- log10(melted_clin$Abundance)

heatCS <- ggplot(melted_clin, aes(Individual, Genus, fill = log10Abundance)) +
  geom_tile() + scale_x_discrete(position = "top") +
  labs(x = "Clinician Swab", fill = "log10 Abundance") +
  theme(axis.title.x = element_text(family = "Helvetica", size = 10, face = "plain"),
        axis.title.y = element_blank(), axis.text = element_blank(),
        legend.position = "bottom", legend.background = element_blank(),
        plot.margin = margin(t = 1, r = 1, b = 0, l = 2, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_gradient(low = "white", high = "red") +
  guides(fill = guide_colourbar(title.position = "bottom", title.hjust = 0.5))

# Self Swab
heat_self <- subset_samples(heat_ps, Sample_type == "Rectal swab ST")
melted_self <- psmelt(heat_self)
melted_self <- select(melted_self, Individual, Genus, Abundance)
melted_self$Abundance[melted_self$Abundance == 0] <- 1
melted_self$log2Abundance <- log2(melted_self$Abundance)
melted_self$log10Abundance <- log10(melted_self$Abundance)

heatSS <- ggplot(melted_self, aes(Individual, Genus, fill = log10Abundance)) +
  geom_tile() + scale_x_discrete(position = "top") +
  labs(x = "Self Swab", fill = "log10 Abundance") +
  theme(axis.title.x = element_text(family = "Helvetica", size = 10, face = "plain"),
        axis.title.y = element_blank(), axis.ticks.y = element_blank(),
        axis.text = element_blank(),
        legend.position = "bottom", legend.background = element_blank(),
        plot.margin = margin(t = 1, r = 1, b = 0, l = 1, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_gradient(low = "white", high = "red") +
  guides(fill = guide_colourbar(title.position = "bottom", title.hjust = 0.5))

# Stool
heat_stool <- subset_samples(heat_ps, Sample_type == "Stool")
melted_stool <- psmelt(heat_stool)
melted_stool <- select(melted_stool, Individual, Genus, Abundance)
```

```
melted_stool$Abundance[melted_stool$Abundance == 0] <- 1
melted_stool$log2Abundance <- log2(melted_stool$Abundance)
melted_stool$log10Abundance <- log10(melted_stool$Abundance)

heatSt <- ggplot(melted_stool, aes(Individual, Genus, fill = log10Abundance)) +
  geom_tile() + scale_x_discrete(position = "top") +
  labs(x = "Stool", fill = "log10 Abundance") +
  theme(axis.title.x = element_text(family = "Helvetica", size = 10, face = "plain"),
        axis.title.y = element_blank(), axis.ticks.y = element_blank(),
        axis.text = element_blank(),
        legend.position = "bottom", legend.background = element_blank(),
        plot.margin = margin(t = 1, r = 1, b = 0, l = 1, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_gradient(low = "white", high = "red") +
  guides(fill = guide_colourbar(title.position = "bottom", title.hjust = 0.5))


heatAbundance <- ggarrange(heatCS, heatSS, heatSt, ncol = 3,
                           common.legend = TRUE, legend = c("bottom"))
```

Combined Heatmaps

```
ggarrange(heatChanges, heatAbundance, widths = c(2, 1), legend = c("bottom"))
```



```
ggsave("../Results/4)Differential_Abundance_heatmap_extra.pdf",
       width = 11, height = 8)
```

Supplementary 2 - Calculate some stats for deseq2 enrichment

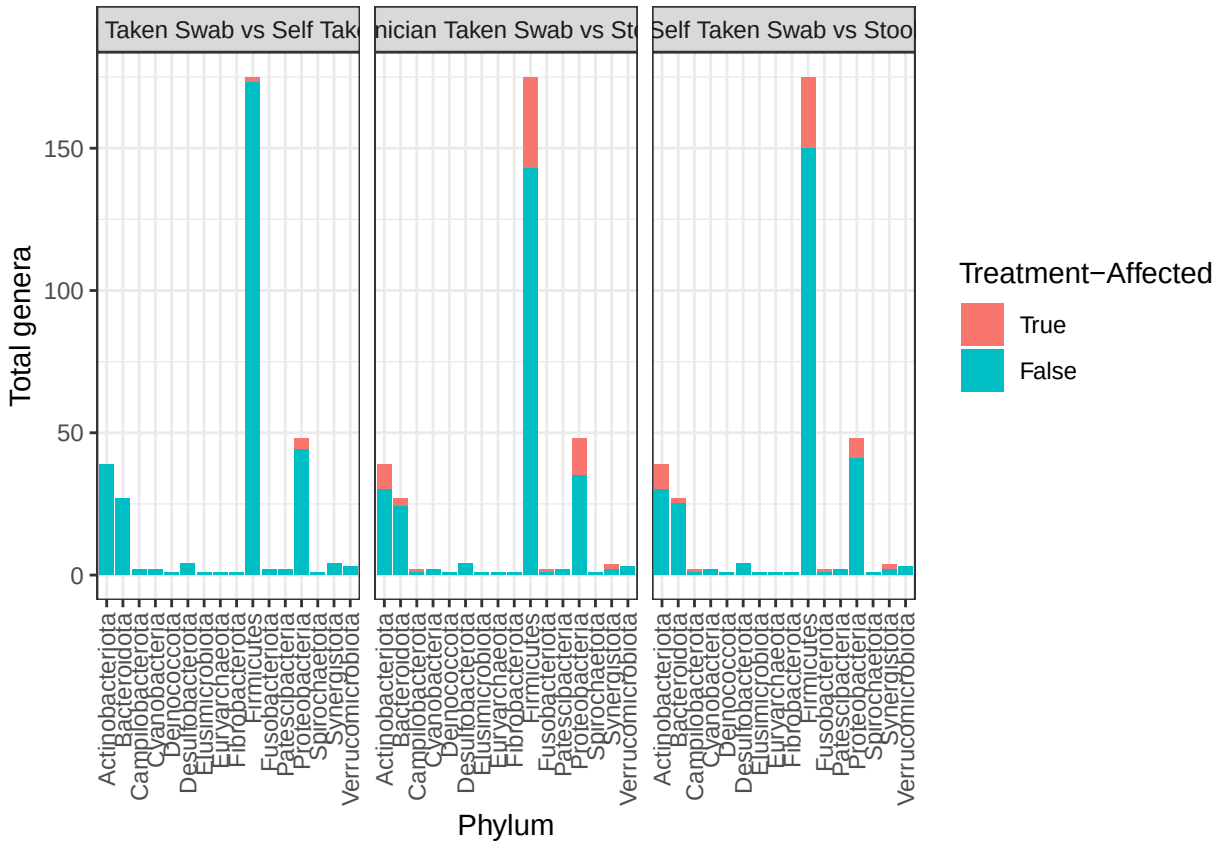
```
mytax <- data.frame(tax_table(ps_deseq))
mytable <- mytax %>%
  group_by(Phylum) %>%
  summarize(Phycount = n_distinct(Genus))
sigdat <- SignificantResults %>%
  group_by(Comparison, Phylum) %>%
  summarize(count = n_distinct(Genus))
```

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

```
mysum <- sum(mytable$Phycount)
#mytable$PhyRatio = mytable$Phycount /mysum
mytable$ConditionA = "Clinician Taken Swab vs Stool"
mytable$ConditionB = "Self Taken Swab vs Stool"
mytable$ConditionC = "Clinician Taken Swab vs Self Taken Swab"

data_long <- tidyr::gather(mytable, condition, Comparison, ConditionA:ConditionC, factor_key = TRUE)
joined2 <- sigdat %>%
  full_join(data_long)
```

```
## Joining, by = c("Comparison", "Phylum")
```

```
joined2<-joined2 %>%
  mutate_at(c(3), ~replace(., is.na(.), 0))
joined2$Phycount = joined2$Phycount - joined2$count

dat_long <- joined2 %>%
    tidyr::gather("count", "measurement", count, Phycount)
dat_long$measurement = as.numeric(as.character(dat_long$measurement))

ggplot(dat_long) +
  geom_col(aes(x = Phylum, y = measurement, fill = count)) +
  facet_grid(~Comparison) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylab("Total genera") +
  scale_fill_discrete(name = "Treatment-Affected", labels = c("True", "False"))
```




```
ggsave("../Results/S2)TreatmentAffectedGenera.pdf", width = 10, height = 6)


joined2$total = joined2$count + joined2$Phycount
joined2$condition = gsub("ConditionC", 0.019, joined2$condition)
joined2$condition = gsub("ConditionB", 0.15, joined2$condition)
joined2$condition = gsub("ConditionA", 0.195, joined2$condition)
joined2$condition = as.numeric(as.character(joined2$condition))

f<-NULL;
for(i in c(1:48))
{
#print(joined2$total[i])
r<-binom.test(joined2$count[i], joined2$total[i], joined2$condition[i])
#print(r)
f<-rbind(f, c(r$statistic, r$parameter, r$null.value, r$p.value))
++i
}
colnames(f)[4] = "p.val"
# Not significant after FDR correction
p.adjust(f[,4], method = "BH")
```

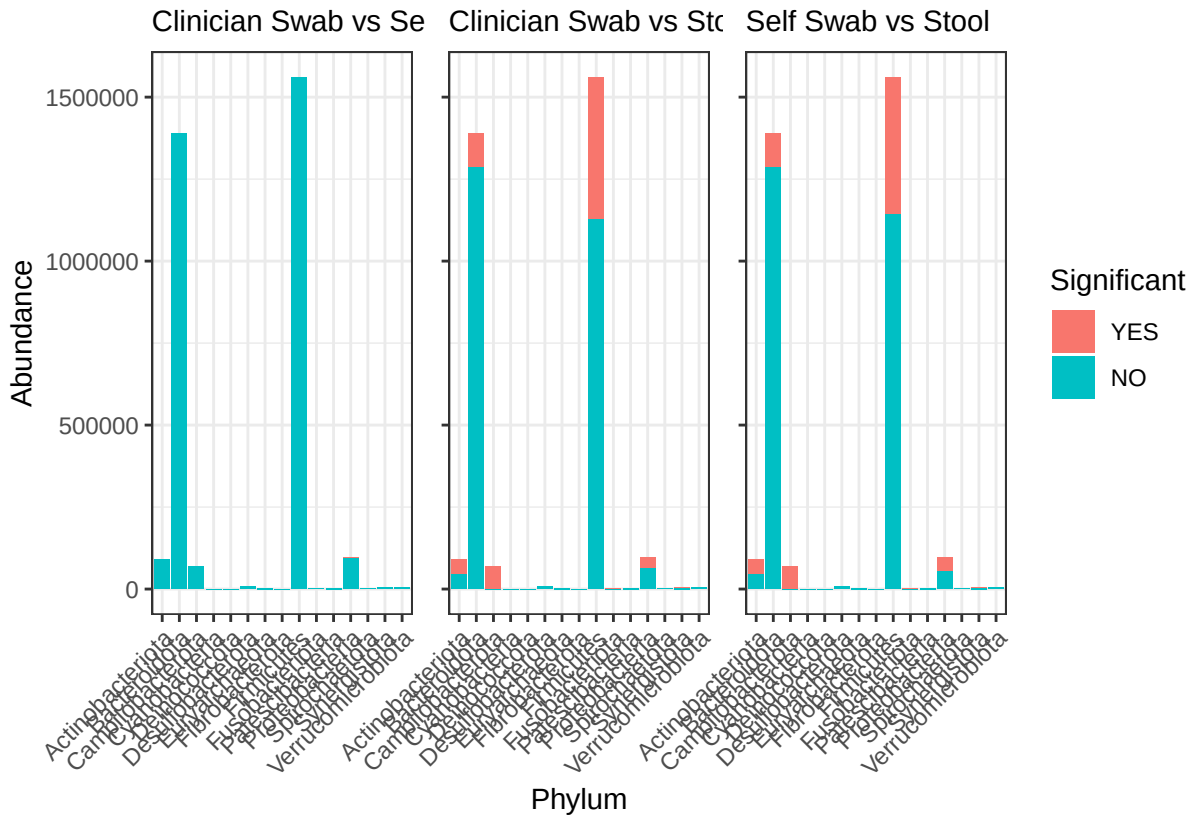
```
##  [1] 1.000000 0.626522 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
##  [9] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
## [17] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
## [25] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
## [33] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
```

```
## [41] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
```

```
joined2[7:10] = f[,1:4]
colnames(joined2)[7:10] = c("successes", "trials", "pSuccess", "pval")
joined2$p.adjust = p.adjust(f[,4], method = "BH")
```

Supplementary 3 - DESeq2 Significance by Abundance

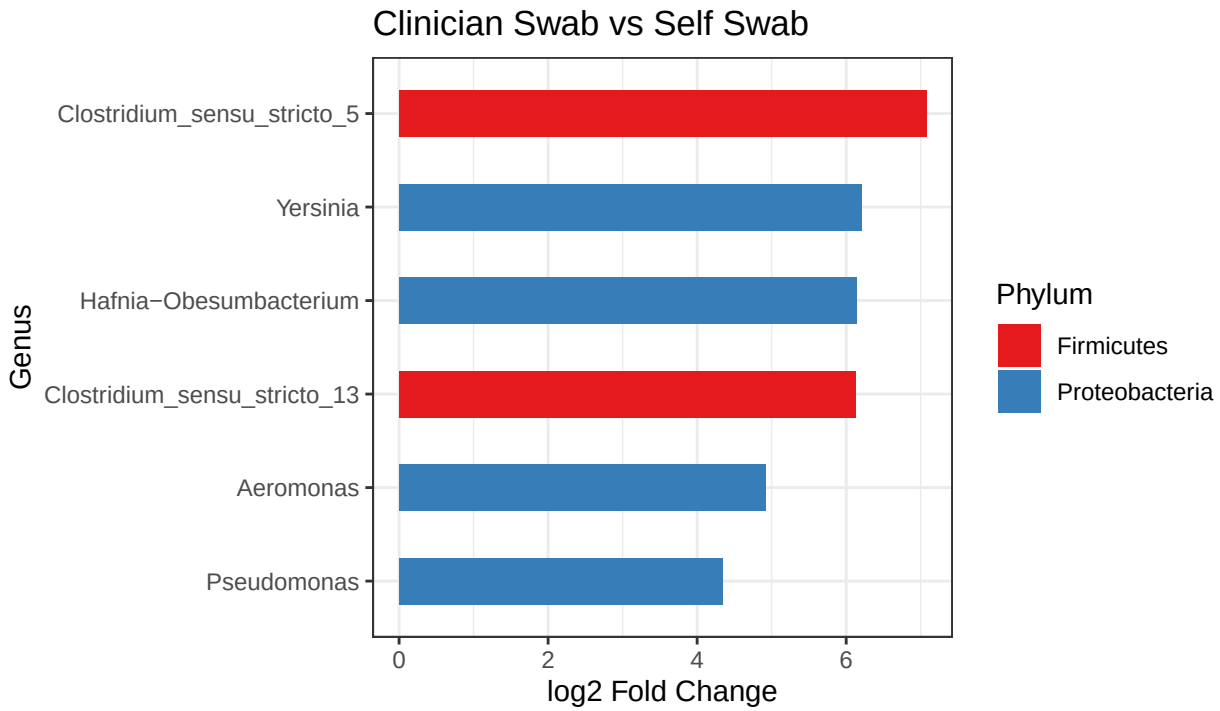
```
library(patchwork)

sup_ps <- ps_rare %>%
  tax_glom(taxrank = "Genus")

sup_bugs <- as.character(unique(SignificantResults$Genus))
sup_bugsCTST <- as.character(resCTST_sig$Genus)
sup_bugsCTS <- as.character(resCTS_sig$Genus)
sup_bugsSTS <- as.character(resSTS_sig$Genus)

sup_melt <- psmelt(sup_ps)
sup_melt$Phylum <- as.character(sup_melt$Phylum)
sup_melt$Genus <- as.character(sup_melt$Genus)

sup_melt$Significant <- ifelse(sup_melt$Genus %in% sup_bugs, "YES", "NO")
sup_melt$Significant <- factor(sup_melt$Significant, levels = c("YES", "NO"))

sup_melt$CTST <- ifelse(sup_melt$Genus %in% sup_bugsCTST, "YES", "NO")
sup_melt$CTST <- factor(sup_melt$CTST, levels = c("YES", "NO"))

sup_melt$CTS <- ifelse(sup_melt$Genus %in% sup_bugsCTS, "YES", "NO")
sup_melt$CTS <- factor(sup_melt$CTS, levels = c("YES", "NO"))

sup_melt$STS <- ifelse(sup_melt$Genus %in% sup_bugsSTS, "YES", "NO")
sup_melt$STS <- factor(sup_melt$STS, levels = c("YES", "NO"))

sup_CTST <- ggplot(sup_melt, aes(x = Phylum, y = Abundance, fill = CTST)) +
  geom_col() + labs(subtitle = "Clinician Swab vs Self Swab", fill = "Significant") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.x = element_blank(), legend.position = "none")

sup_CTS <- ggplot(sup_melt, aes(x = Phylum, y = Abundance, fill = CTS)) +
  geom_col() + labs(subtitle = "Clinician Swab vs Stool", fill = "Significant") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_blank(), axis.text.y = element_blank(),
        legend.position = "none")

sup_STS <- ggplot(sup_melt, aes(x = Phylum, y = Abundance, fill = STS)) +
  geom_col() + labs(subtitle = "Self Swab vs Stool", fill = "Significant") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title.x = element_blank(),
        axis.title.y = element_blank(), axis.text.y = element_blank())

sup_CTST + sup_CTS + sup_STS + plot_layout(ncol = 3)
```



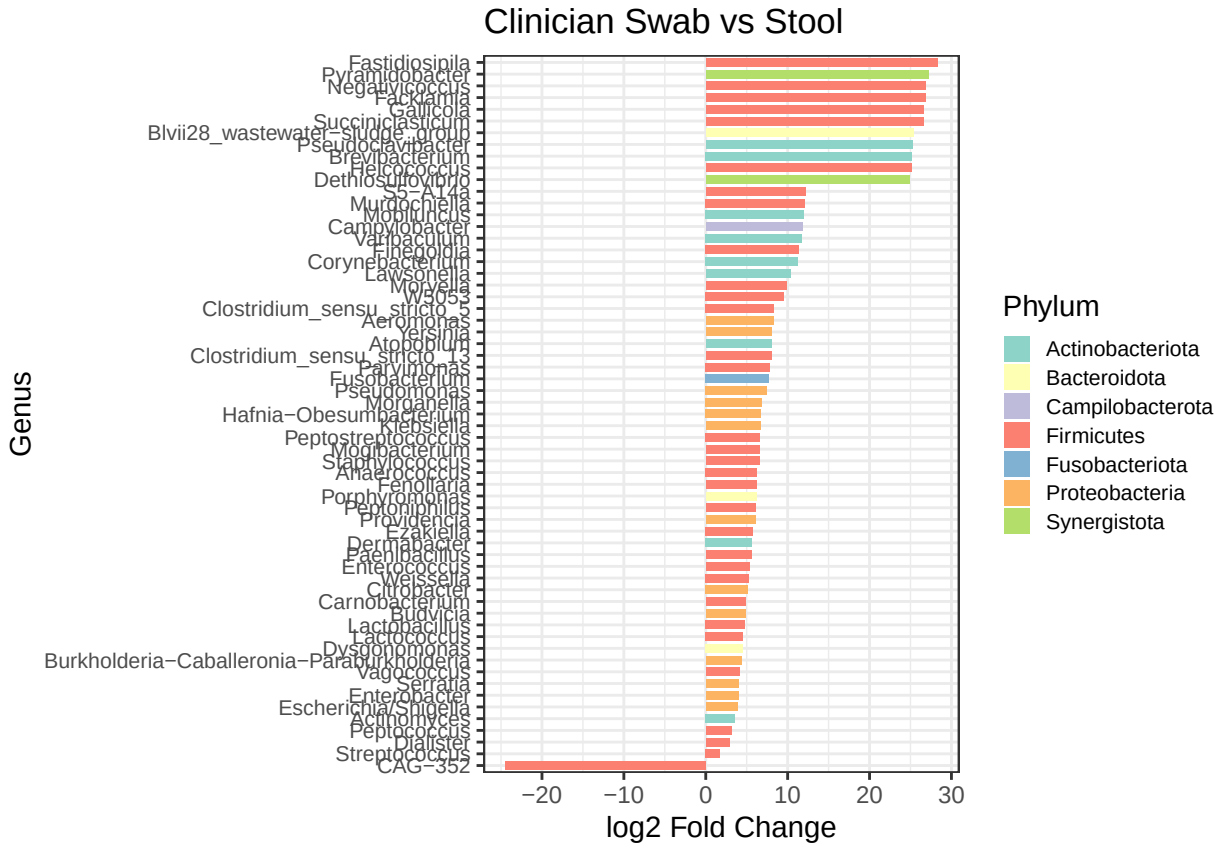

```
ggsave("../Results/S3)SignificanceByAbundance.pdf", width = 10, height = 6)
```

Supplementary 4 - Differential Abundance Figure

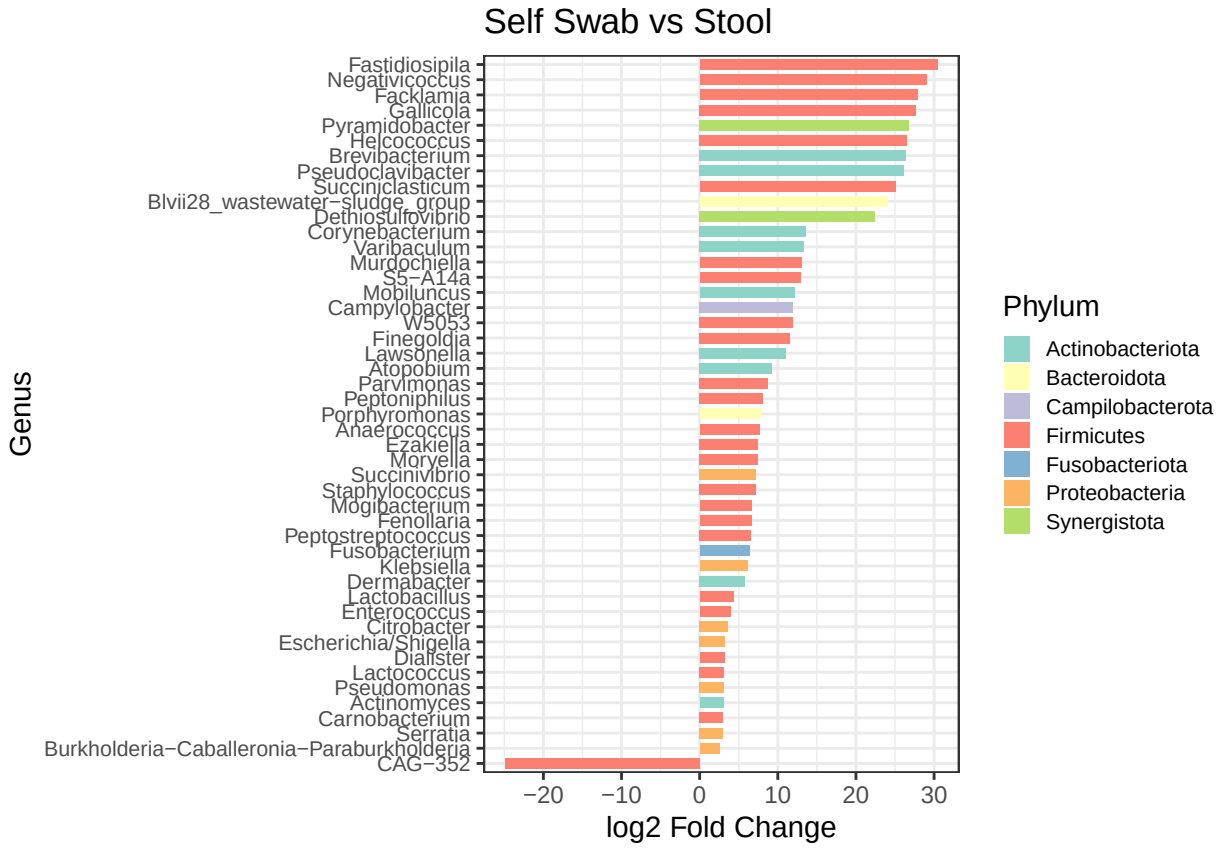
```
ggplot(resCTST_sig, aes(x = log2FoldChange, y = reorder(Genus, log2FoldChange), fill= Phylum)) +
  geom_bar(stat = "identity", position = "identity", width = 0.5) +
  labs(title = "Clinician Swab vs Self Swab", y = "Genus", x = "log2 Fold Change") +
  theme(aspect.ratio = 1) +
  scale_fill_brewer(palette = "Set1")
```



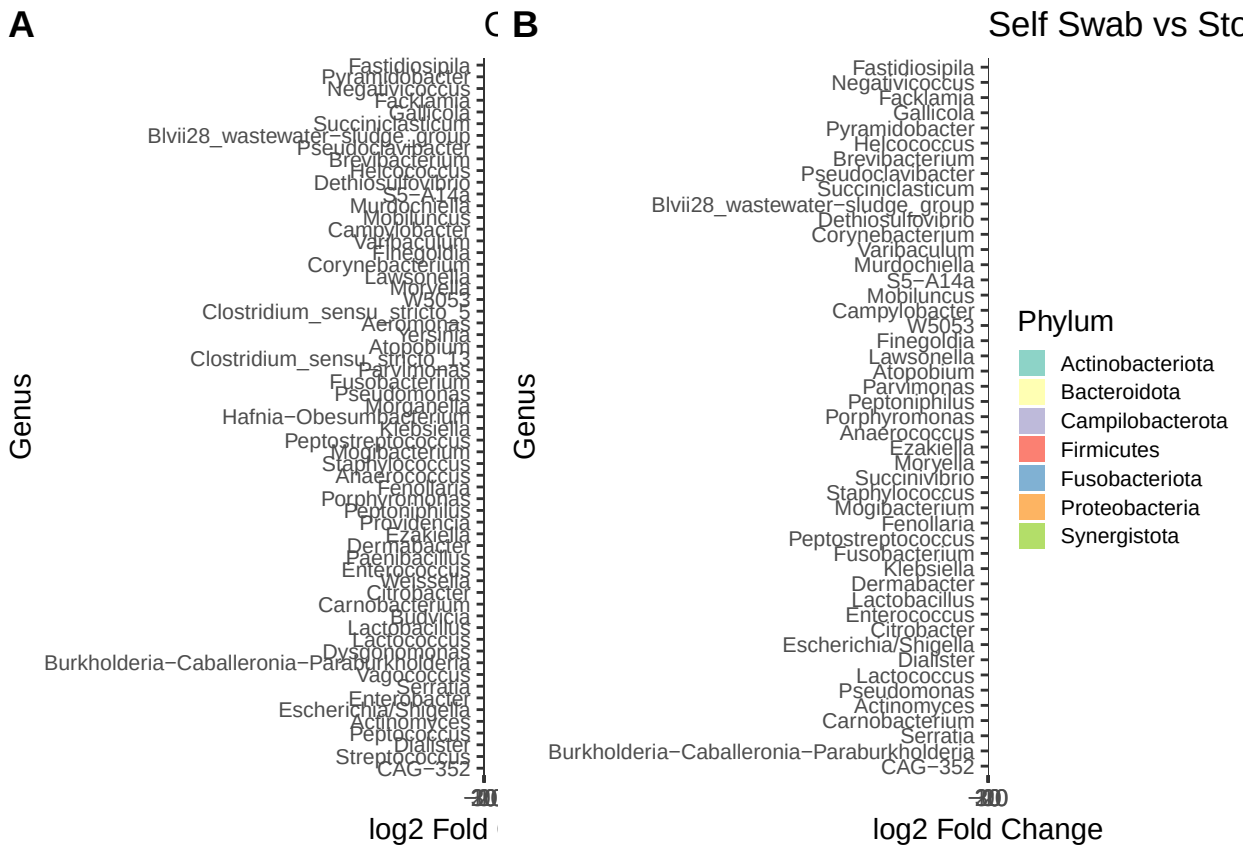

```
ggsave("../Results/S4)Differential_Abundance_clinVSself.pdf", width = 7, height = 4)

clinVSstool <- ggplot(resCTS_sig, aes(x = log2FoldChange,
                                      y = reorder(Genus, log2FoldChange),
                                      fill= Phylum)) +
  geom_bar(stat = "identity", position = "identity", width = 0.7) +
  labs(title = "Clinician Swab vs Stool", y = "Genus", x = "log2 Fold Change") +
  scale_fill_brewer(palette = "Set3") +
  theme(axis.text.y = element_text(size = 8),
        legend.text = element_text(size = 8), legend.key.size = unit(0.75, "line"))
clinVSstool
```




```
selfVSstool <- ggplot(resSTS_sig, aes(x = log2FoldChange,
                                      y = reorder(Genus, log2FoldChange),
                                      fill= Phylum)) +
  geom_bar(stat = "identity", position = "identity", width = 0.7) +
  labs(title = "Self Swab vs Stool", y = "Genus", x = "log2 Fold Change") +
  scale_fill_brewer(palette = "Set3") +
  theme(axis.text.y = element_text(size = 8),
        legend.text = element_text(size = 8), legend.key.size = unit(0.75, "line"))
selfVSstool
```




```
ggarrange(clinVSstool, selfVSstool, ncol = 2, common.legend = TRUE, legend = "right", labels = "AUTO")
```




```
ggsave("../Results/S4)Differential_Abundance_swabsVSstool.pdf", width = 12, height = 8)
```

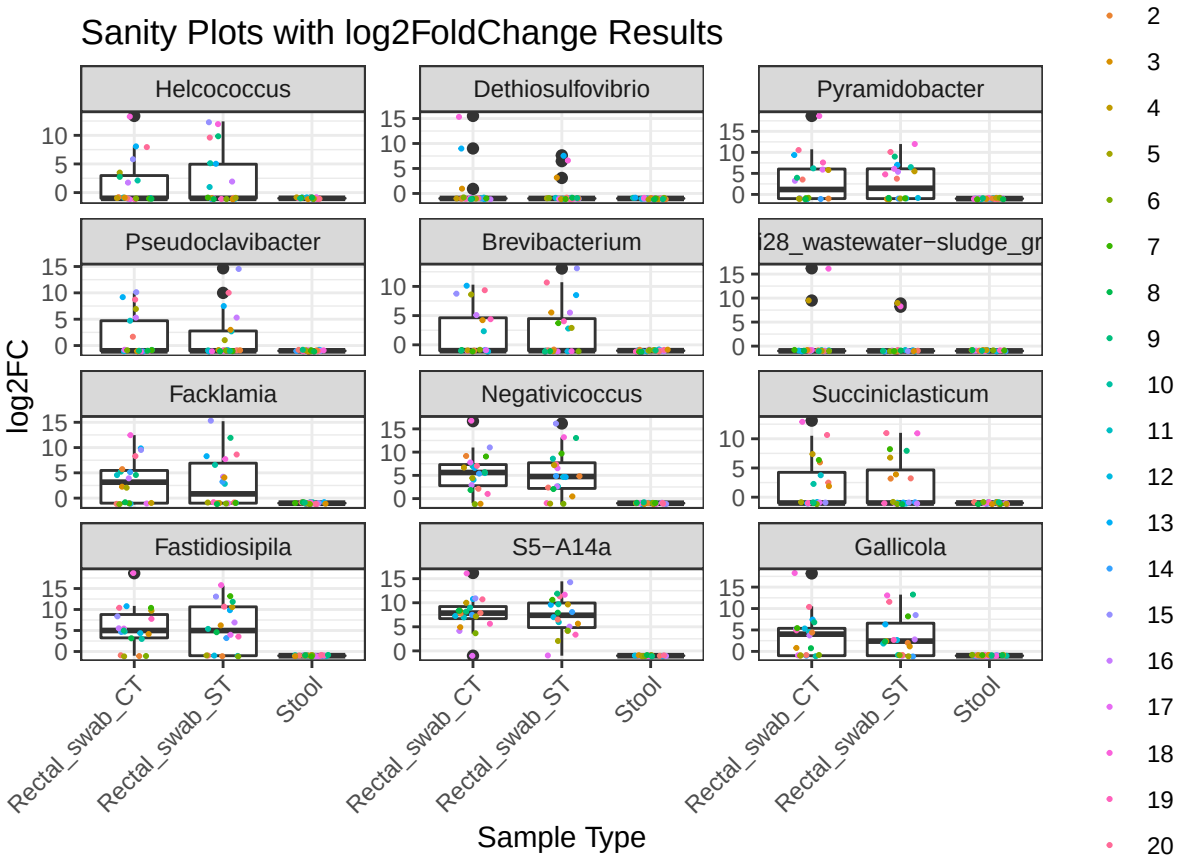
Supplementary 5 - Boxplot Sanity Checks

```
resCTS_sig <- resCTS_sig[order(-resCTS_sig$log2FoldChange),]

int <- row.names(resCTS_sig)[1:12]
ASVlabs <- tax_table(ps)[int, 6]
names(ASVlabs) <- int
ASVlabs <- as.list(ASVlabs)

ASV_labeller <- function(variable,value){
  return(ASVlabs[value])
}

# Sanity Plots with Fold Change
tcounts <- t(log2((counts(ds[int, ], normalized = TRUE, replaced = FALSE) + .5))) %>%
  merge(colData(ds), ., by = "row.names") %>%
  tidyr::gather(ASV, log2FC, (ncol(.)-length(int) + 1):ncol(.))

tcounts %>%
  select(Row.names, Sample_type, Individual, ASV, log2FC) %>%
  head %>%
  knitr::kable()
```

Row.names	Sample_type	Individual	ASV	log2FC
10A	Rectal_swab_CT	10	ASV3599	3.017179
10B	Rectal_swab_ST	10	ASV3599	5.359164
10C	Stool	10	ASV3599	-1.000000
11A	Rectal_swab_CT	11	ASV3599	4.888552
11B	Rectal_swab_ST	11	ASV3599	-1.000000
11C	Stool	11	ASV3599	-1.000000

```
ggplot(tcounts, aes(Sample_type, log2FC)) +
  geom_boxplot() + geom_jitter(width = 0.2, height = 0.2, size = 0.4, aes(color = Individual)) +
  facet_wrap(~ASV, scales = "free_y", labeller = ASV_labeller, nrow = 4) +
  labs(x = "Sample Type",
       y = "log2FC",
       title = "Sanity Plots with log2FoldChange Results") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: The labeller API has been updated. Labellers taking `variable` and
## `value` arguments are now deprecated. See labellers documentation.
```



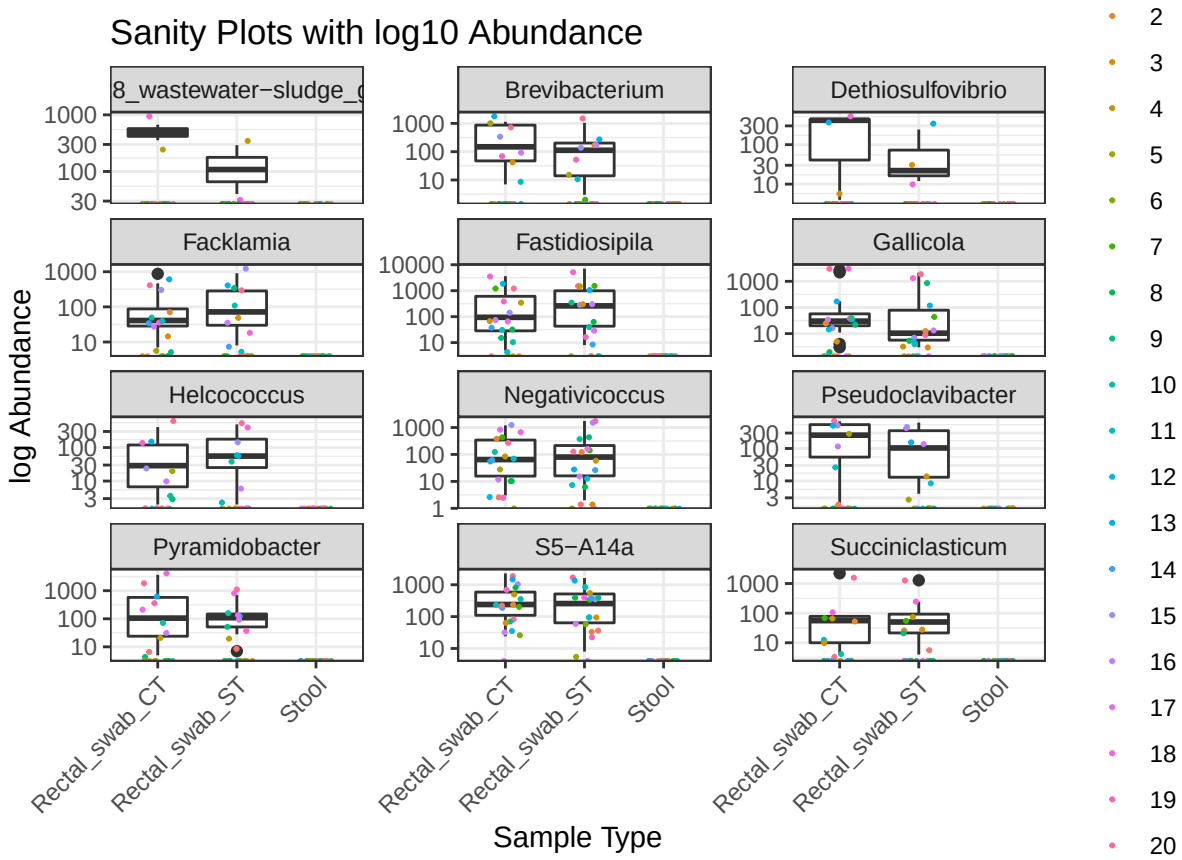

```
ggsave("../Results/S5)Sanity_FoldChange_plots.pdf", width = 7, height = 8)

# Sanity Plots with Abundance
sanity_ps <- subset_taxa(ps_deseq, taxa_names(ps_deseq) %in% int)
sanity <- psmelt(sanity_ps)

ggplot(sanity, aes(Sample_type, Abundance)) +
```

```
  geom_boxplot() + geom_jitter(width = 0.2, height = 0.2, size = 0.4, aes(color = Individual)) +
  facet_wrap(~Genus, scales = "free_y", nrow = 4) +
  scale_y_log10() +
  labs(x = "Sample Type",
       y = "log Abundance",
       title = "Sanity Plots with log10 Abundance") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 474 rows containing non-finite values (stat_boxplot).
```




```
ggsave("../Results/S5)Sanity_logAbundance_plots.pdf", width = 7, height = 8)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 474 rows containing non-finite values (stat_boxplot).
```

Session Info

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
```

```
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_NZ.UTF-8/en_NZ.UTF-8/en_NZ.UTF-8/C/en_NZ.UTF-8/en_NZ.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] patchwork_1.1.1             DESeq2_1.30.1
##  [3] SummarizedExperiment_1.20.0 Biobase_2.50.0
##  [5] MatrixGenerics_1.2.1        matrixStats_0.58.0
##  [7] GenomicRanges_1.42.0        GenomeInfoDb_1.26.7
##  [9] IRanges_2.24.1              S4Vectors_0.28.1
## [11] BiocGenerics_0.36.1         ggpubr_0.4.0
## [13] ggplot2_3.3.3               phyloseq_1.34.0
## [15] dplyr_1.0.5                 vegan_2.5-7
## [17] permute_0.9-5               Rmisc_1.5
## [19] plyr_1.8.6                  lattice_0.20-41
## [21] RColorBrewer_1.1-2
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-0       ggsignif_0.6.1         ellipsis_0.3.1
##   [4] rio_0.5.26             XVector_0.30.0         farver_2.1.0
##   [7] bit64_4.0.5            AnnotationDbi_1.52.0   fansi_0.4.2
##  [10] codetools_0.2-18       splines_4.0.3          cachem_1.0.4
##  [13] geneplotter_1.68.0     knitr_1.32             ade4_1.7-16
##  [16] jsonlite_1.7.2         broom_0.7.6            annotate_1.68.0
##  [19] cluster_2.1.1          compiler_4.0.3         httr_1.4.2
##  [22] backports_1.2.1        fastmap_1.1.0          assertthat_0.2.1
##  [25] Matrix_1.3-2           htmltools_0.5.1.1      prettyunits_1.1.1
##  [28] tools_4.0.3            igraph_1.2.6           gtable_0.3.0
##  [31] glue_1.4.2             GenomeInfoDbData_1.2.4 reshape2_1.4.4
##  [34] Rcpp_1.0.6             carData_3.0-4          cellranger_1.1.0
##  [37] vctrs_0.3.7            Biostrings_2.58.0      rhdf5filters_1.2.0
##  [40] multtest_2.46.0        ape_5.4-1              nlme_3.1-152
##  [43] iterators_1.0.13       xfun_0.22              stringr_1.4.0
##  [46] openxlsx_4.2.3         lifecycle_1.0.0        rstatix_0.7.0
##  [49] XML_3.99-0.6           zlibbioc_1.36.0        MASS_7.3-53.1
##  [52] scales_1.1.1           hms_1.0.0              biomformat_1.18.0
##  [55] rhdf5_2.34.0           yaml_2.2.1             curl_4.3
##  [58] gridExtra_2.3          memoise_2.0.0          stringi_1.5.3
##  [61] RSQLite_2.2.5          highr_0.9              genefilter_1.72.1
##  [64] foreach_1.5.1          zip_2.1.1              BiocParallel_1.24.1
##  [67] rlang_0.4.10           pkgconfig_2.0.3        bitops_1.0-6
##  [70] evaluate_0.14          purrr_0.3.4            Rhdf5lib_1.12.1
##  [73] labeling_0.4.2         cowplot_1.1.1          bit_4.0.4
##  [76] tidyselect_1.1.0       magrittr_2.0.1         R6_2.5.0
##  [79] generics_0.1.0         DelayedArray_0.16.3    DBI_1.1.1
```

```
##  [82] pillar_1.6.0          haven_2.4.0           foreign_0.8-81
##  [85] withr_2.4.2           mgcv_1.8-34           survival_3.2-10
##  [88] abind_1.4-5           RCurl_1.98-1.3        tibble_3.1.1
##  [91] crayon_1.4.1          car_3.0-10            utf8_1.2.1
##  [94] rmarkdown_2.7         progress_1.2.2        locfit_1.5-9.4
##  [97] grid_4.0.3            readxl_1.3.1          data.table_1.14.0
## [100] blob_1.2.1            forcats_0.5.1         digest_0.6.27
## [103] xtable_1.8-4          tidyr_1.1.3           munsell_0.5.0
```