Purcell Project Markdown

Dan Hudson

07/August/2020

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("data/ps_notree.rds")

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

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

# 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 ]
## refseq()      DNAStringSet:      [ 4872 reference sequences ]
```

Custom Rarefaction Plot

Not run in this Markdown

```
# Data
psdata <- ps

# Loading required library and displaying core configuration
```

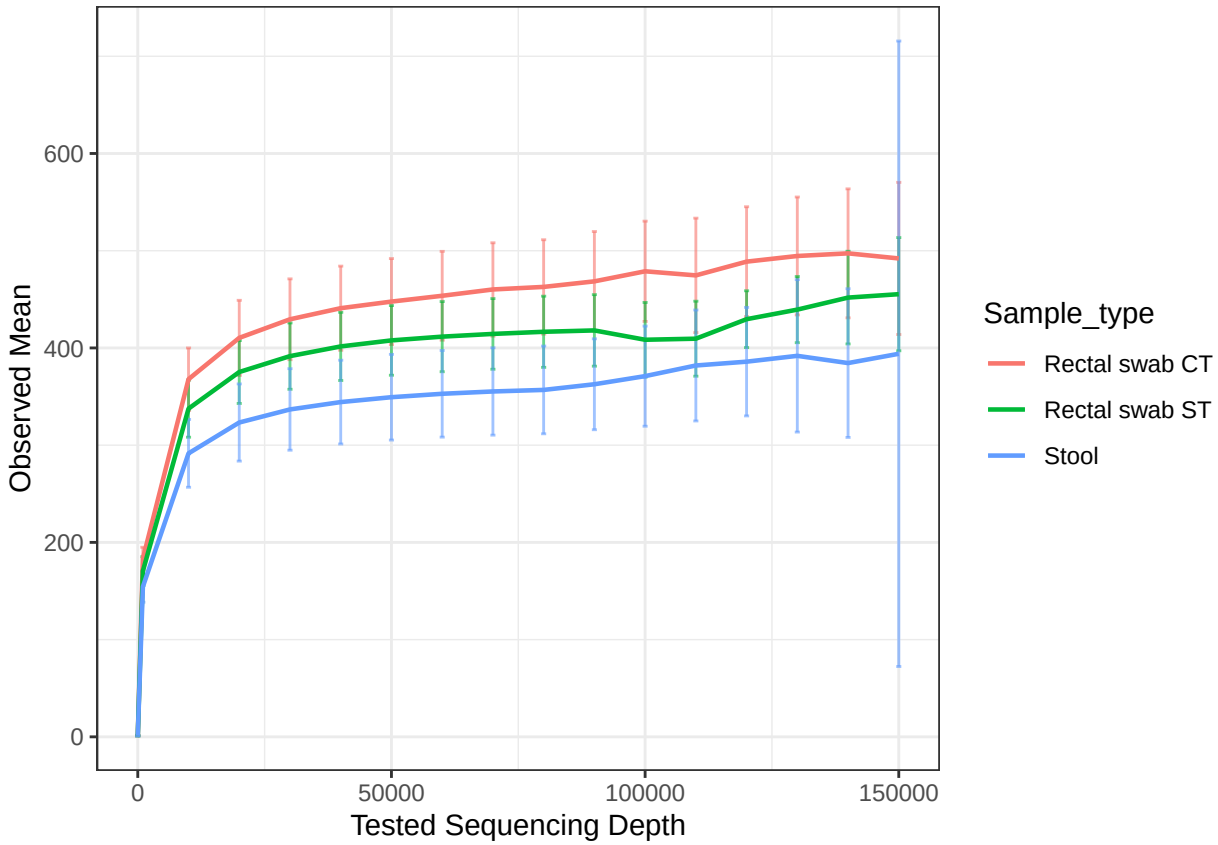
```r
library('doParallel')
detectCores(all.tests = TRUE)

# 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, 1:150 * 1000),
                                                        each = 10))
summary(rarefaction_curve_data)
saveRDS(rarefaction_curve_data, file = "Final_results/rare_object.rds")
```

```
# Data
psdata <- ps

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

```
##      Depth            Sample          Measure      Alpha_diversity
##  Min.   :     1   X10B   : 1520   Observed:77740   Min.   :  1.0
##  1st Qu.: 31000   X12B   : 1520                    1st Qu.:321.0
##  Median : 63000   X12C   : 1520                    Median :403.0
##  Mean   : 65150   X13A   : 1520                    Mean   :391.3
##  3rd Qu.: 97000   X13B   : 1520                    3rd Qu.:464.0
##  Max.   :150000   X14A   : 1520                    Max.   :674.0
##                   (Other):68620
```

```
# 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

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 = 1000, alpha = 0.6) +
  geom_line(size = 0.8) +
  ylab("Observed Mean") +
  xlab("Tested Sequencing Depth")
```




```
ggsave("Purcell Final/Final_results/1)Rarefaction_Curve.pdf", width = 11, height = 8)
```

Rarefy

```
# Rarefy to even sequencing depth, 90% of minimum sample depth
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

## ...

## 2520TUs 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"))
shapiro.test(alpha_summary$Observed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha_summary$Observed
## W = 0.99236, p-value = 0.971
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha_summary$Shannon
## W = 0.97837, p-value = 0.3634
```

```
# 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)
```

```r
summary(avO)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## tm           2 106371   53186  14.343 2.29e-05 ***
## blk         19 334512   17606   4.748 2.22e-05 ***
## Residuals   38 140911    3708
## ---
## 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   6.550 0.003596 **
## blk         19  4.828  0.2541   3.919 0.000167 ***
## Residuals   38  2.464  0.0648
## ---
## 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 0.1362         -
## Stool          0.0023         0.0680
##
## P value adjustment method: BH
```

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

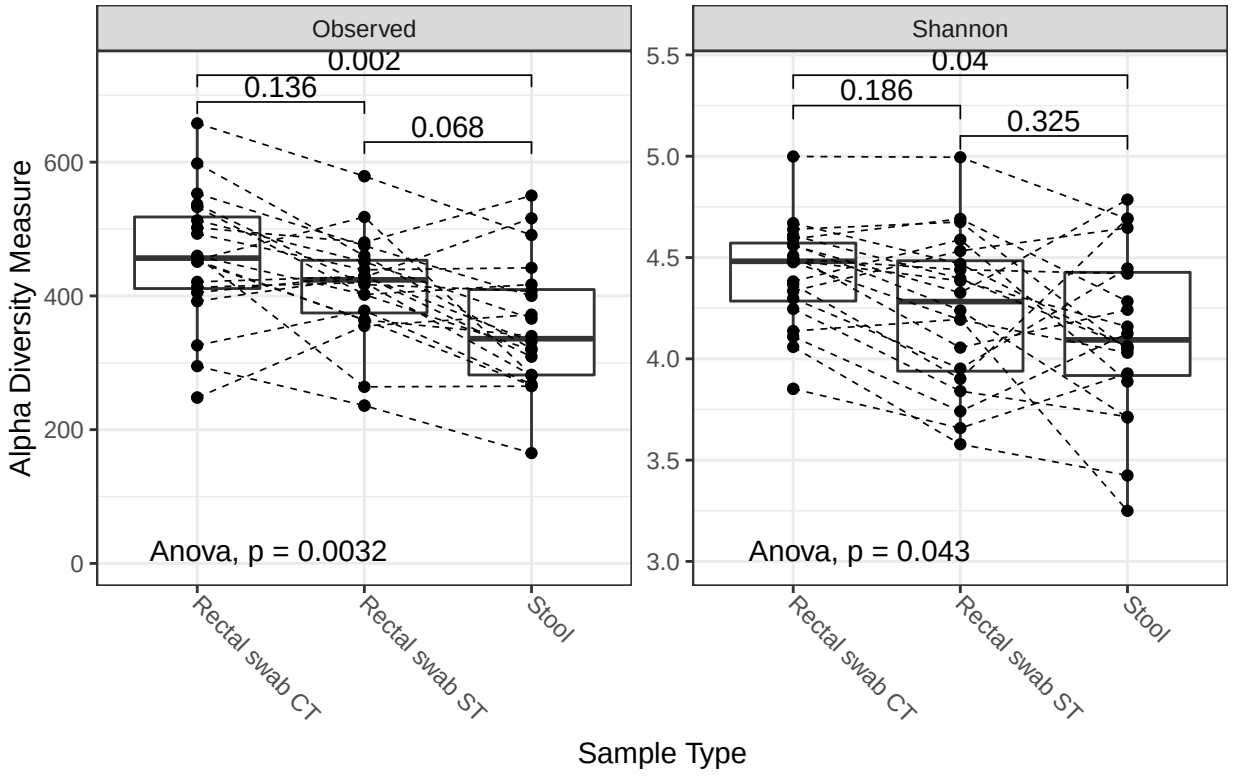
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  alpha_summary$Shannon and sample_data(ps_rare)$Sample_type
##
##                Rectal swab CT Rectal swab ST
## Rectal swab ST 0.19           -
## Stool          0.04           0.32
##
## P value adjustment method: BH
```

```r
# 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), size = 0.3, linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 315, hjust = 0),
        aspect.ratio = 1, legend.position = "none") +
  stat_pvalue_manual(pAdjusted) +
  stat_compare_means(method = "anova", label.y = 3)
```



```r
ggsave("Purcell Final/Final_results/2)Alpha_Diversity.pdf", width = 7, height = 4.5)
```

Beta Diversity - Bray-Curtis

```r
# 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")
```

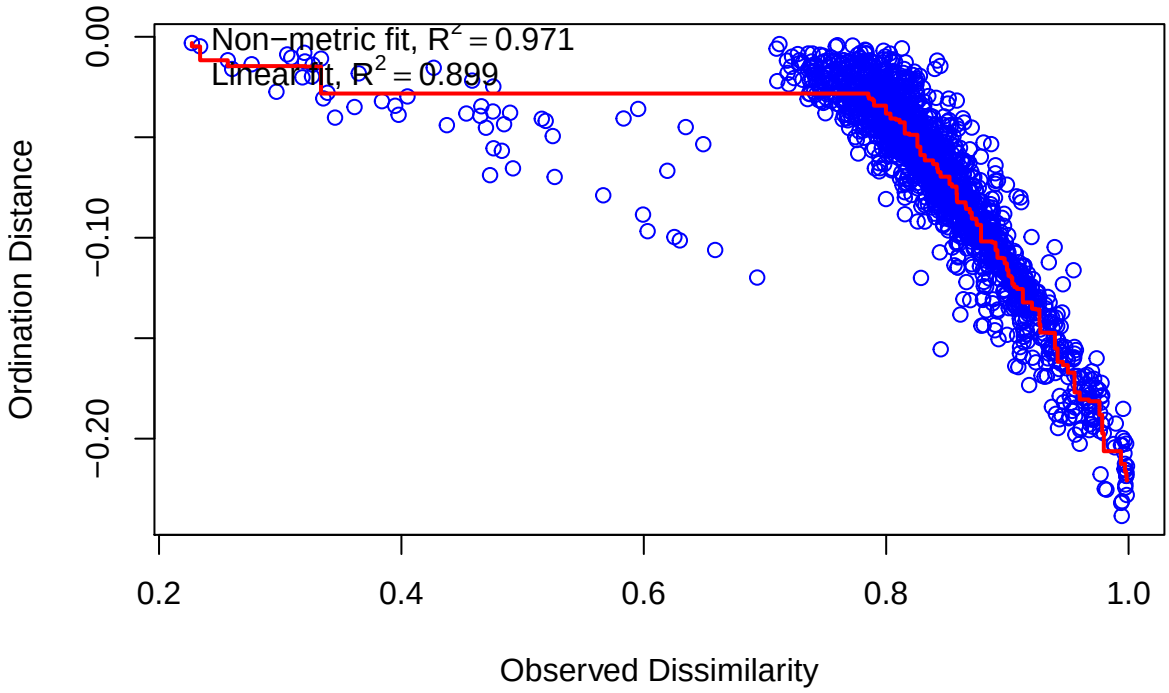
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1715739
## Run 1 stress 0.1691834
## ... New best solution
## ... Procrustes: rmse 0.04497538  max resid 0.2003563
## Run 2 stress 0.1713637
## Run 3 stress 0.1713643
## Run 4 stress 0.1691834
## ... New best solution
## ... Procrustes: rmse 3.069457e-05  max resid 0.0001884229
## ... Similar to previous best
## Run 5 stress 0.1776206
## Run 6 stress 0.1691834
## ... Procrustes: rmse 4.190845e-05  max resid 0.0002593319
## ... Similar to previous best
## Run 7 stress 0.1691917
## ... Procrustes: rmse 0.005989015  max resid 0.03671881
## Run 8 stress 0.1691834
## ... Procrustes: rmse 5.879927e-05  max resid 0.0002700656
## ... Similar to previous best
## Run 9 stress 0.171576
## Run 10 stress 0.1691151
## ... New best solution
## ... Procrustes: rmse 0.003849059  max resid 0.02310026
## Run 11 stress 0.1691156
## ... Procrustes: rmse 0.0001153643  max resid 0.0006062887
## ... Similar to previous best
## Run 12 stress 0.2130362
## Run 13 stress 0.2041349
## Run 14 stress 0.1691152
## ... Procrustes: rmse 4.170052e-05  max resid 0.0001642921
## ... Similar to previous best
## Run 15 stress 0.1972185
## Run 16 stress 0.1691151
## ... Procrustes: rmse 0.000100886  max resid 0.0005413709
## ... Similar to previous best
## Run 17 stress 0.1691152
## ... Procrustes: rmse 0.0001151884  max resid 0.0007715556
## ... Similar to previous best
## Run 18 stress 0.1713635
## Run 19 stress 0.1691846
## ... Procrustes: rmse 0.003872846  max resid 0.02326164
```

```
## Run 20 stress 0.171575
## *** 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:     0.1691151
## 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)$Sample_type  2    1.3632 0.68158  2.0802 0.06802  0.001
## Residuals                        57   18.6763 0.32766         0.93198
## Total                            59   20.0395                 1.00000
##
## sample_data(ps_rare)$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   14.9138 0.78494  6.1256 0.74422  0.001 ***
## Residuals                       40    5.1257 0.12814         0.25578
## Total                           59   20.0395                 1.00000
## ---
## 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)$Sample_type                                 2    1.3632
## sample_data(ps_rare)$Individual                                 19   14.9138
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 38    3.7625
## Residuals                                                        0    0.0000
## Total                                                           59   20.0395
##                                                                 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)$Sample_type                                 0.06802      1
## sample_data(ps_rare)$Individual                                  0.74422      1
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 0.18775      1
## Residuals                                                        0.00000
## Total                                                            1.00000
```

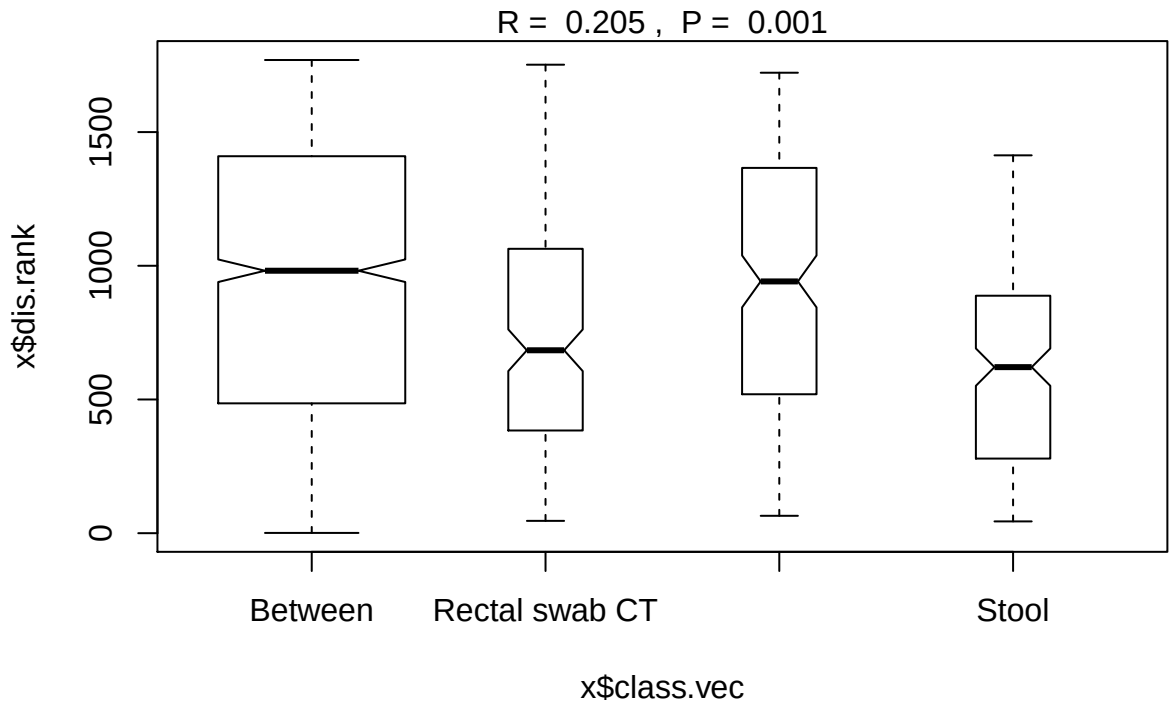
```
anosim(bray_dist, sample_data(ps_rare)$Sample_type)
```

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.2051
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

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

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.2051
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0296 0.0437 0.0568 0.0730
##
## Dissimilarity ranks between and within classes:
##                0%     25%   50%      75%   100%    N
## Between         1 485.750 981.5 1409.125 1769.5 1200
## Rectal swab CT 46 388.500 684.0 1061.875 1752.0  190
## Rectal swab ST 65 519.875 941.5 1365.000 1722.0  190
## Stool          44 279.500 621.0  885.250 1413.0  190
```

```
plot(anoSamp)
```




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

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.7883
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
```

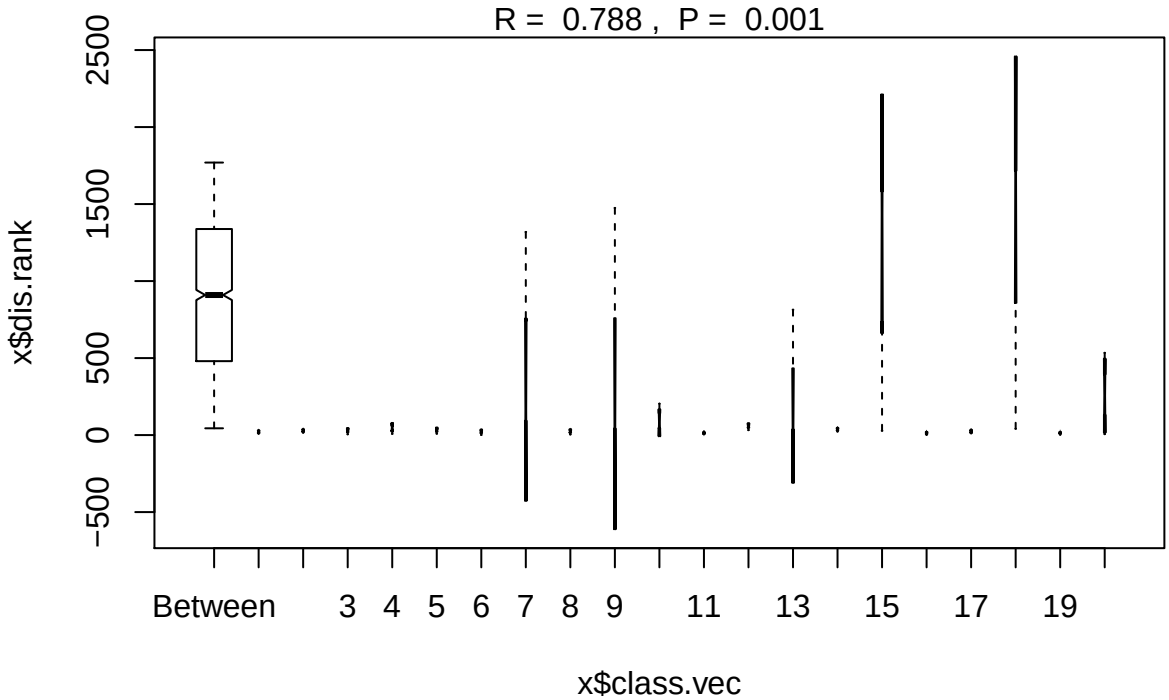
```
anoInd <- anosim(bray_dist, sample_data(ps_rare)$Individual)
summary(anoInd)
```

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Individual)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.7883
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0743 0.1008 0.1253 0.1516
##
## Dissimilarity ranks between and within classes:
```

```
##         0%    25%    50%     75%   100%    N
## Between 44 480.25  909.5 1337.75 1769.5 1710
## 1       10  15.50   21.0   26.00   31.0    3
## 2       16  22.50   29.0   33.00   37.0    3
## 3        5  19.00   33.0   33.50   34.0    3
## 4        8  29.50   51.0   60.50   70.0    3
## 5        9  22.50   36.0   37.00   38.0    3
## 6        1  13.00   25.0   25.50   26.0    3
## 7       20  92.50  165.0  742.00 1319.0    3
## 8        4  15.50   27.0   28.50   30.0    3
## 9       40  41.50   43.0  759.00 1475.0    3
## 10      11  45.50   80.0  142.00  204.0    3
## 11       7  10.00   13.0   17.50   22.0    3
## 12      32  47.50   63.0   63.50   64.0    3
## 13      19  35.50   52.0  433.00  814.0    3
## 14      23  31.00   39.0   40.50   42.0    3
## 15      28 732.50 1437.0 1582.50 1728.0    3
## 16       2   8.00   14.0   16.00   18.0    3
## 17      12  18.00   24.0   29.50   35.0    3
## 18      41 858.50 1676.0 1717.00 1758.0    3
## 19       3   9.00   15.0   16.00   17.0    3
## 20       6 131.00  256.0  395.00  534.0    3
```

```
plot(anoInd)
```

```
## Warning in bxp(list(stats = structure(c(44, 480, 909.5, 1338, 1769.5, 10, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```




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

```
## Analysis of Variance Table
##
```

```
## Response: Distances
##           Df   Sum Sq   Mean Sq F value  Pr(>F)
## Groups     2 0.010524 0.0052620  2.7349 0.07343 .
## Residuals 57 0.109671 0.0019241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(ps.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 0.010524 0.0052620 2.7349    999  0.082 .
## Residuals 57 0.109671 0.0019241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

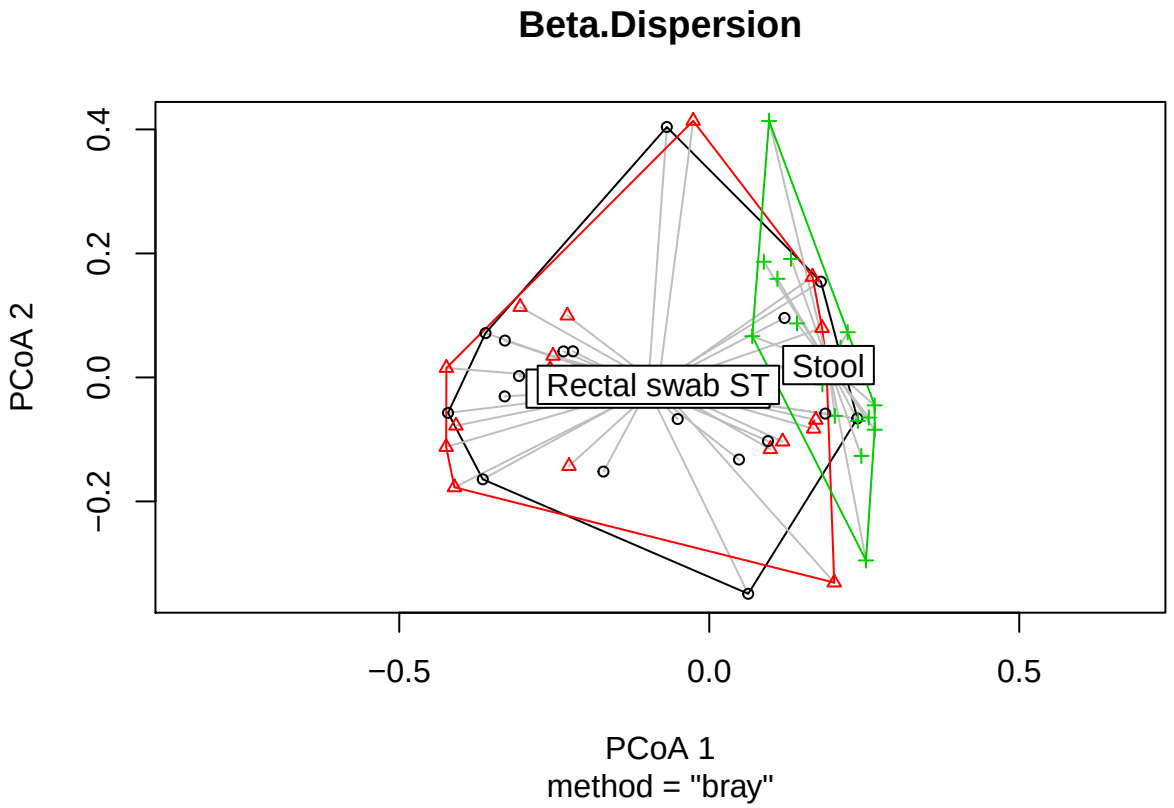
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.010524 0.0052620 2.7349    999  0.082 .
## Residuals 57 0.109671 0.0019241
## ---
## 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                       0.149000 0.399
## Rectal swab ST       0.152212                 0.032
## Stool                0.391940       0.027007
```

```
TukeyHSD(ps.disper)
```

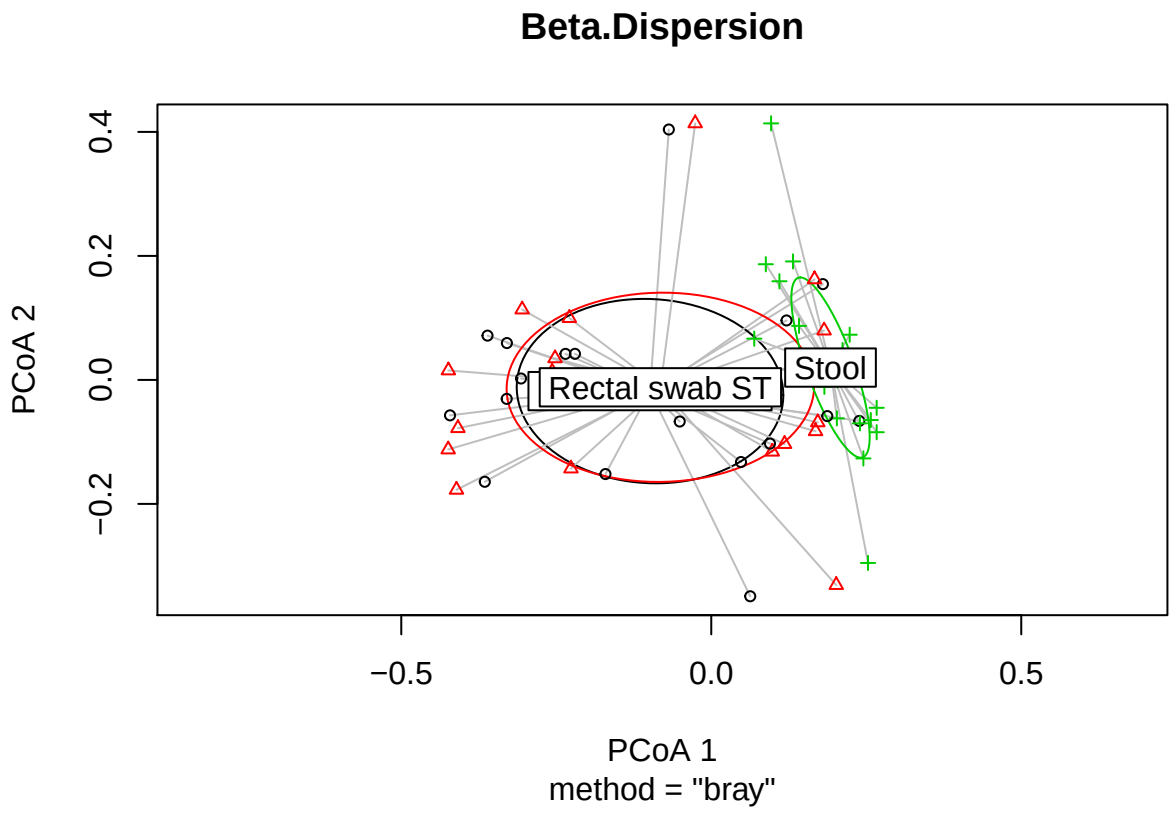
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                     diff         lwr         upr     p adj
## Rectal swab ST-Rectal swab CT  0.02004686 -0.01333259 0.053426322 0.3249206
## Stool-Rectal swab CT          -0.01206488 -0.04544434 0.021314578 0.6613886
## Stool-Rectal swab ST          -0.03211174 -0.06549120 0.001267714 0.0617399
```

```
# Beta Dispersion Plots
Beta.Dispersion <- ps.disper
```

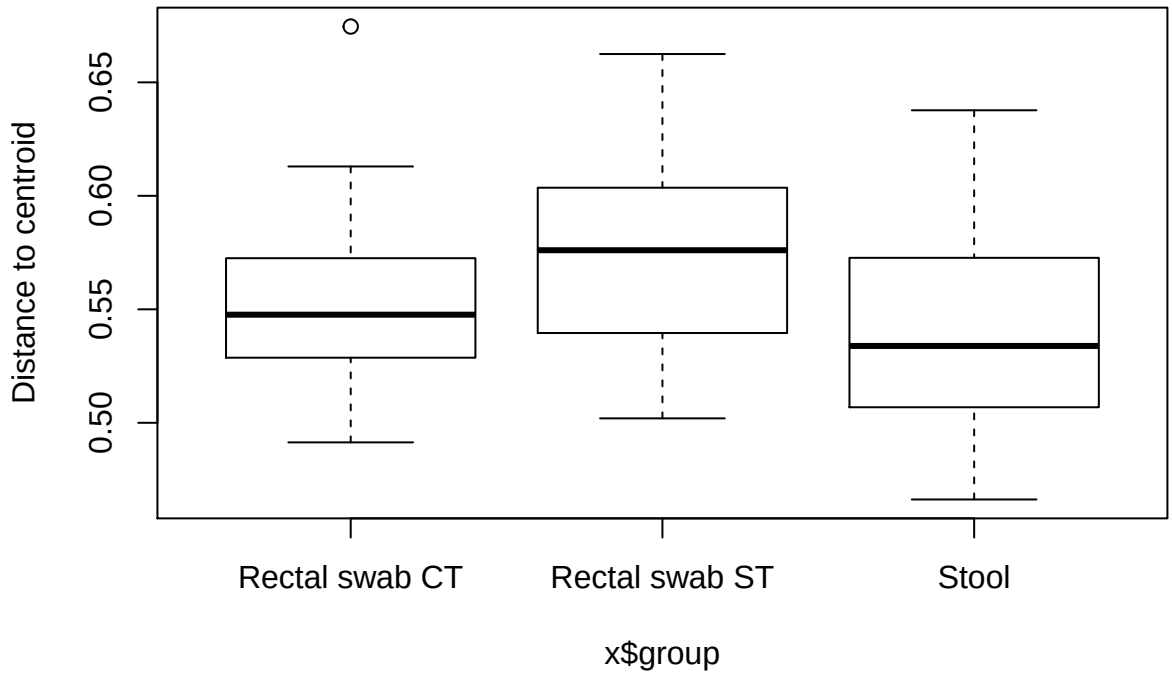
```r
plot(Beta.Dispersion)
```




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

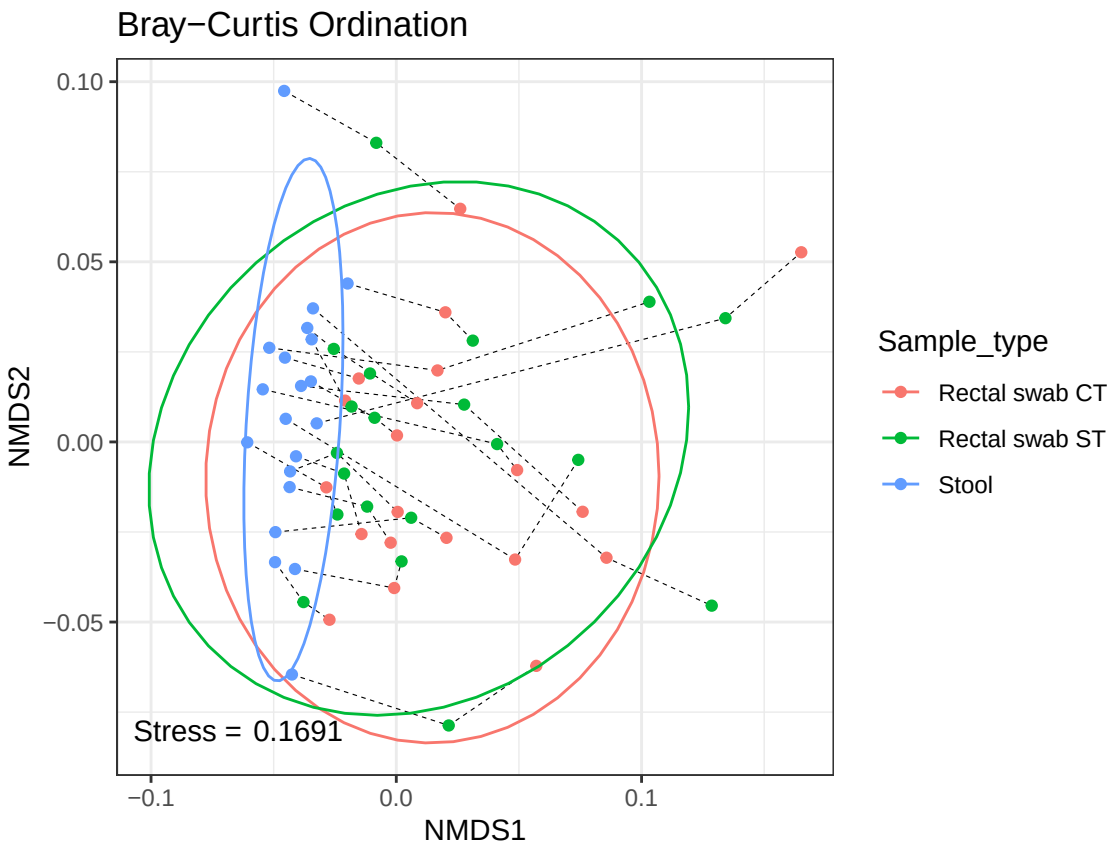



```
boxplot(Beta.Dispersion)
```




```
# NMDS plot
cust <- plot_ordination(ps_rare, ord.nmds.bray, justDF = TRUE)

ggplot(cust, aes(x = NMDS1, y = NMDS2)) +
  geom_line(aes(group = Individual), size = 0.2, linetype = "dashed") +
```

```
  geom_point(aes(color = Sample_type)) +
  annotate("text", x = -0.085, y = -0.08, label = "Stress =") +
  annotate("text", x = -0.04, y = -0.08, label = round(ord.nmds.bray$stress, 4)) +
  stat_ellipse(aes(color = Sample_type)) +
  ggtitle("Bray-Curtis Ordination") +
  theme(aspect.ratio = 1)
```



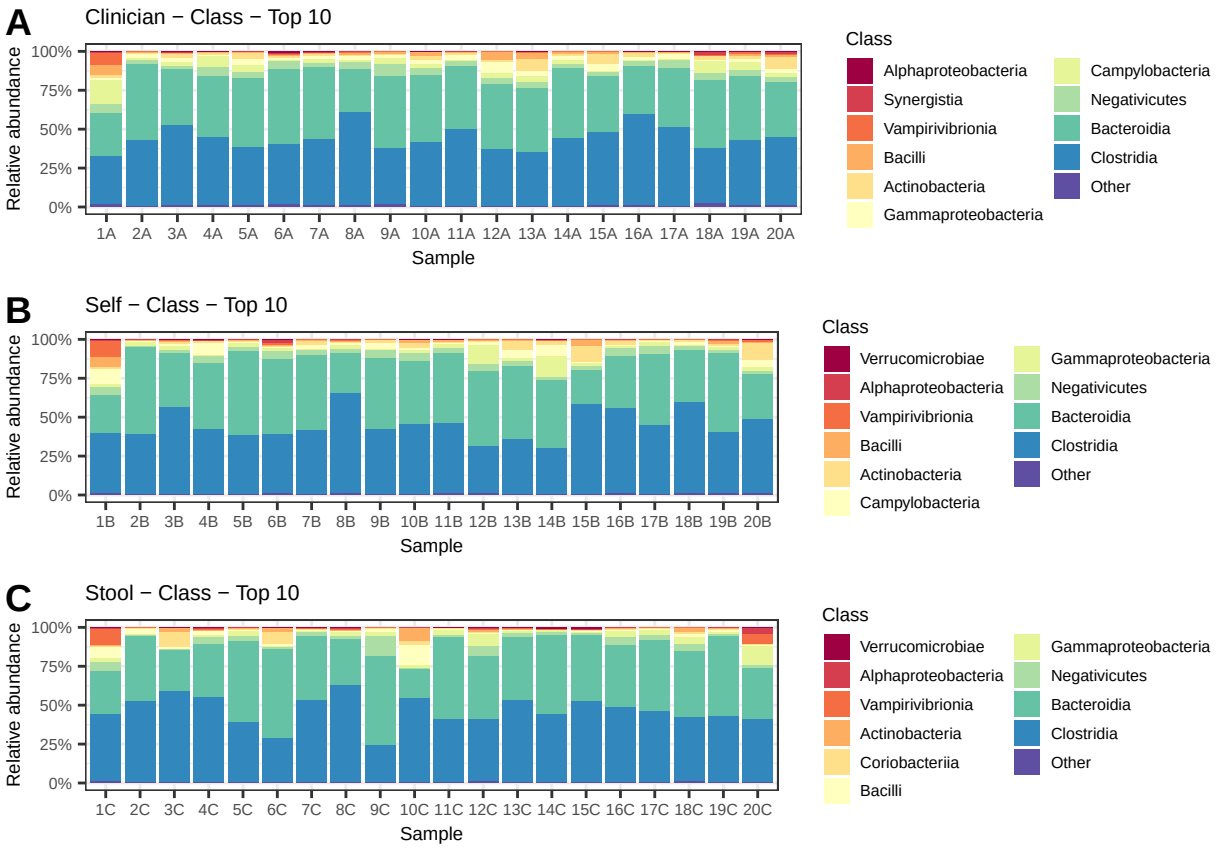

```
ggsave("Purcell Final/Final_results/3)Beta_Diversity.pdf", width = 6, height = 4.5)
```

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_transform <- transform_sample_counts(clin_class, function(x) x/sum(x)) #get abundance in %
clin_melt <- psmelt(clin_transform) # 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"
```

```r
# Set order of bars
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)

# Plot
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(sort.clin))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = percent_format()) +
  theme(text = element_text(size = 7)) +
  ggtitle("Clinician - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_brewer(palette = "Spectral", guide = guide_legend(ncol = 2)) +
  theme(legend.text = element_text(size = 6), legend.key.size = unit(0.75, "line"))


# 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"

# Set order of bars
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)

# Plot
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(sort.self))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = percent_format()) +
```

```
  theme(text = element_text(size = 7)) +
  ggtitle("Self - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_brewer(palette = "Spectral", guide = guide_legend(ncol = 2)) +
  theme(legend.text = element_text(size = 6), legend.key.size = unit(0.75, "line"))


# 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
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)

# Plot
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(sort.stool))) %>%
  ggplot(aes(x = Sample, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = percent_format()) +
  theme(text = element_text(size = 7)) +
  ggtitle("Stool - Class - Top 10") +
  ylab("Relative abundance") +
  scale_fill_brewer(palette = "Spectral", guide = guide_legend(ncol = 2)) +
  theme(legend.text = element_text(size = 6), legend.key.size = unit(0.75, "line"))

ggarrange(t1_class, t2_class, t3_class, nrow = 3, labels = "AUTO", legend = "right")
```




```
ggsave("Purcell Final/Final_results/4)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
```

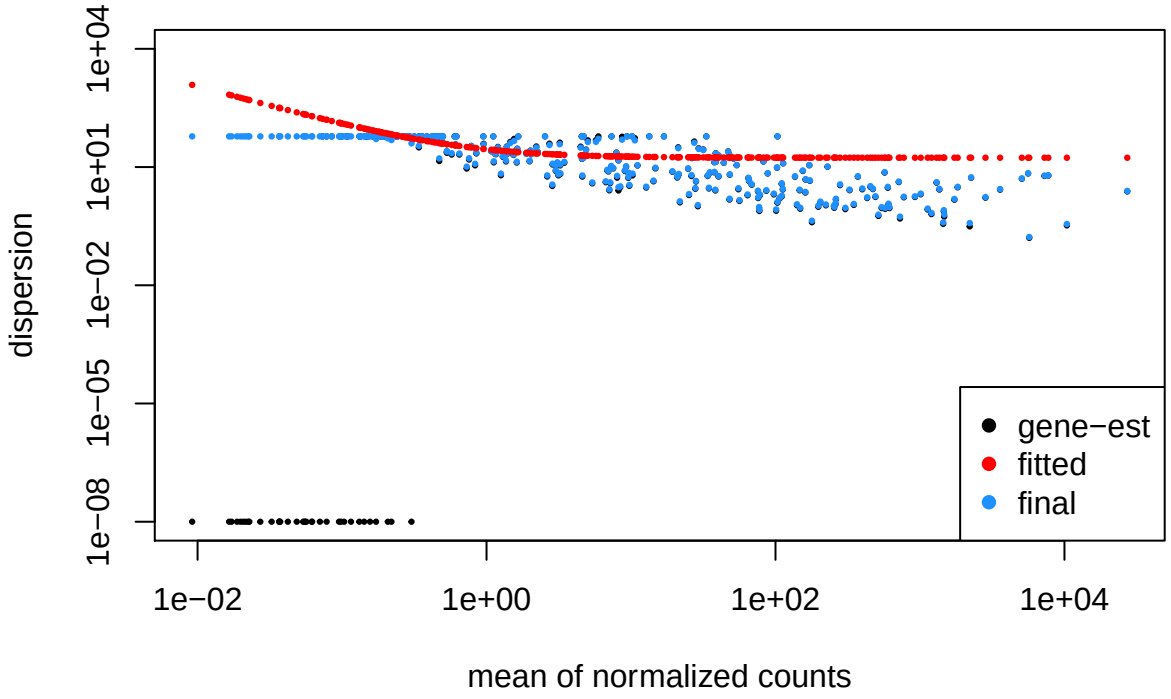
```
## fitting model and testing
```

```
## -- replacing outliers and refitting for 151 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

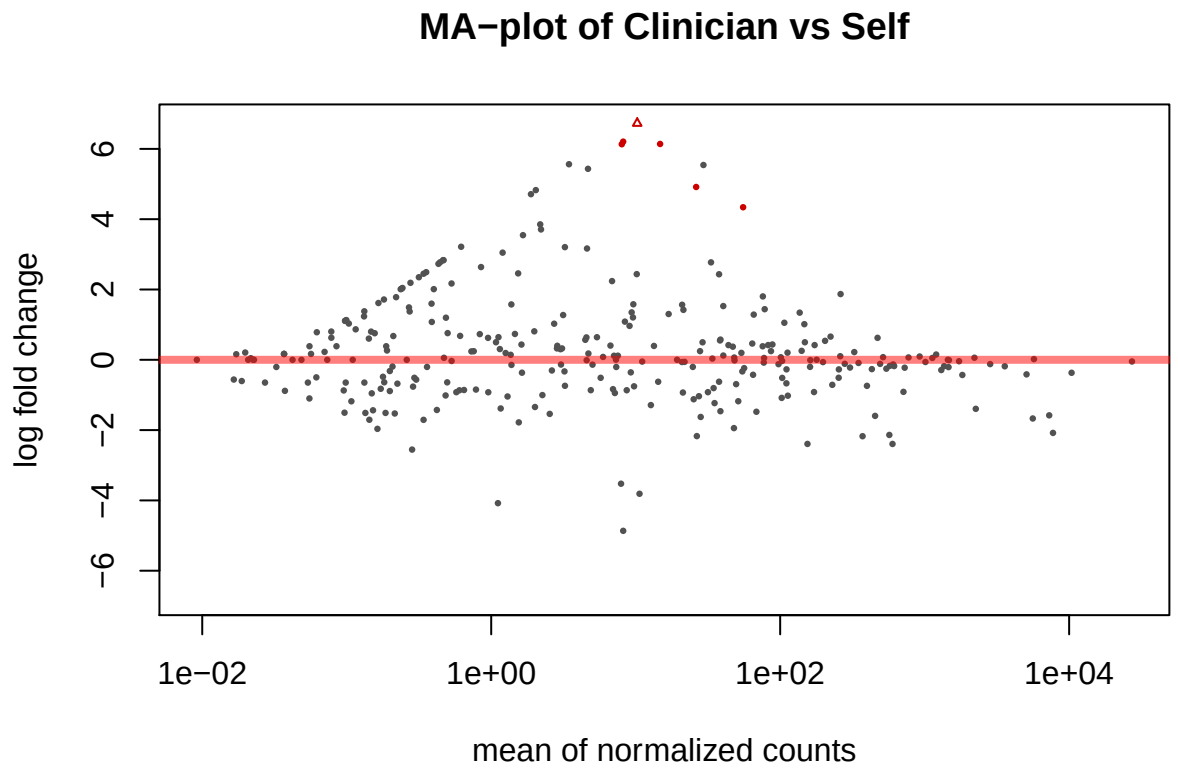
```
## estimating dispersions
```

```
## fitting model and testing
```

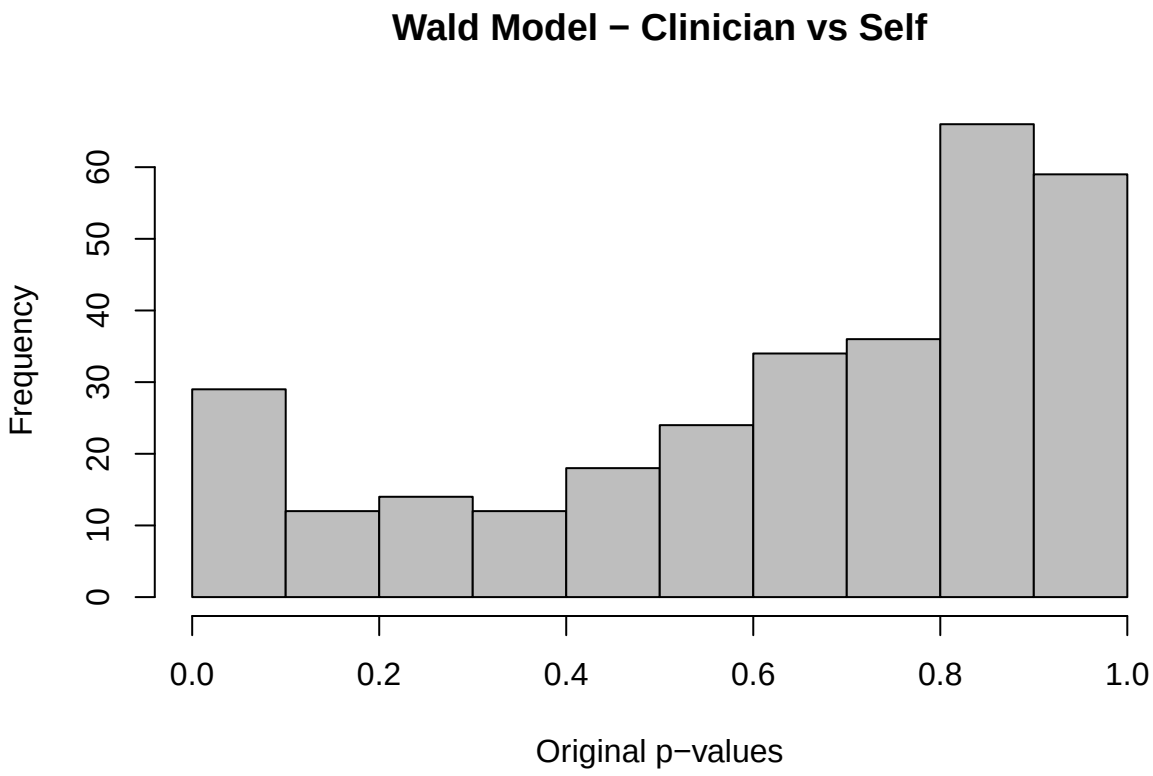
```
# Plot of Dispersion Estimates
plotDispEsts(ds, ylim = c(1e-8, 1e4))
```




```
# 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")
```



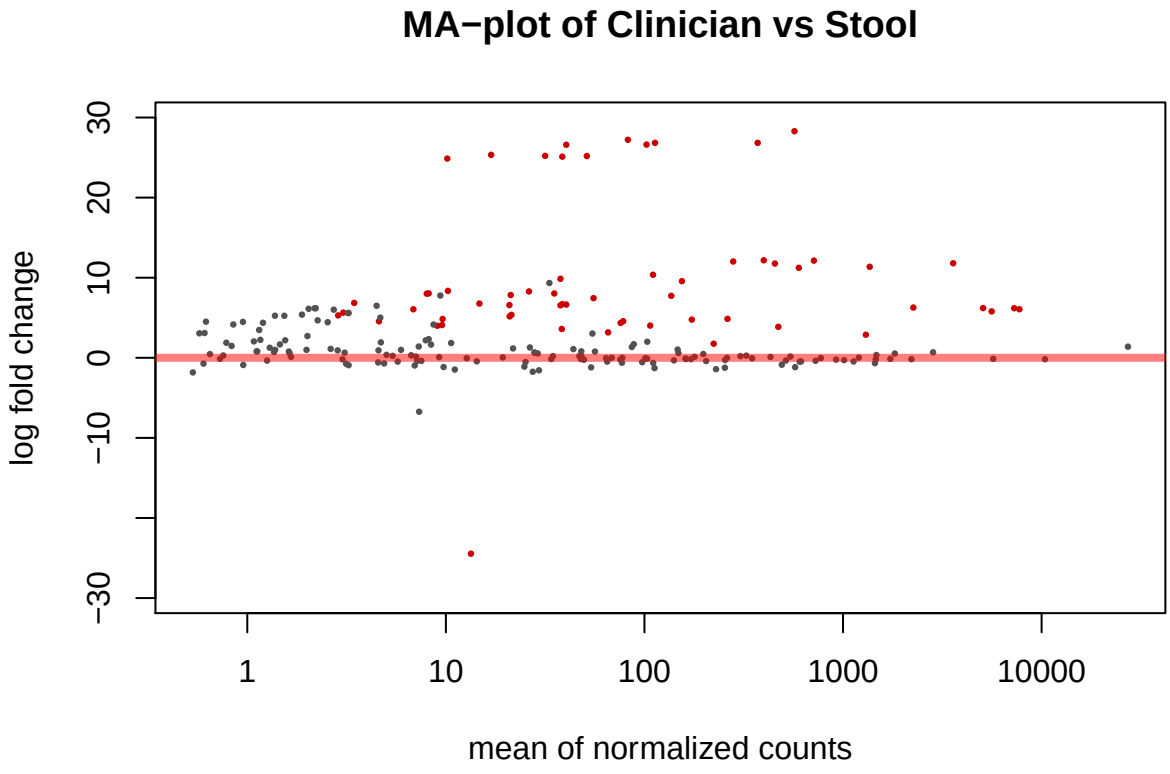

```r
hist(resCTST$pvalue, col = "gray", main = "Wald Model - Clinician vs Self", xlab = "Original p-values")
```



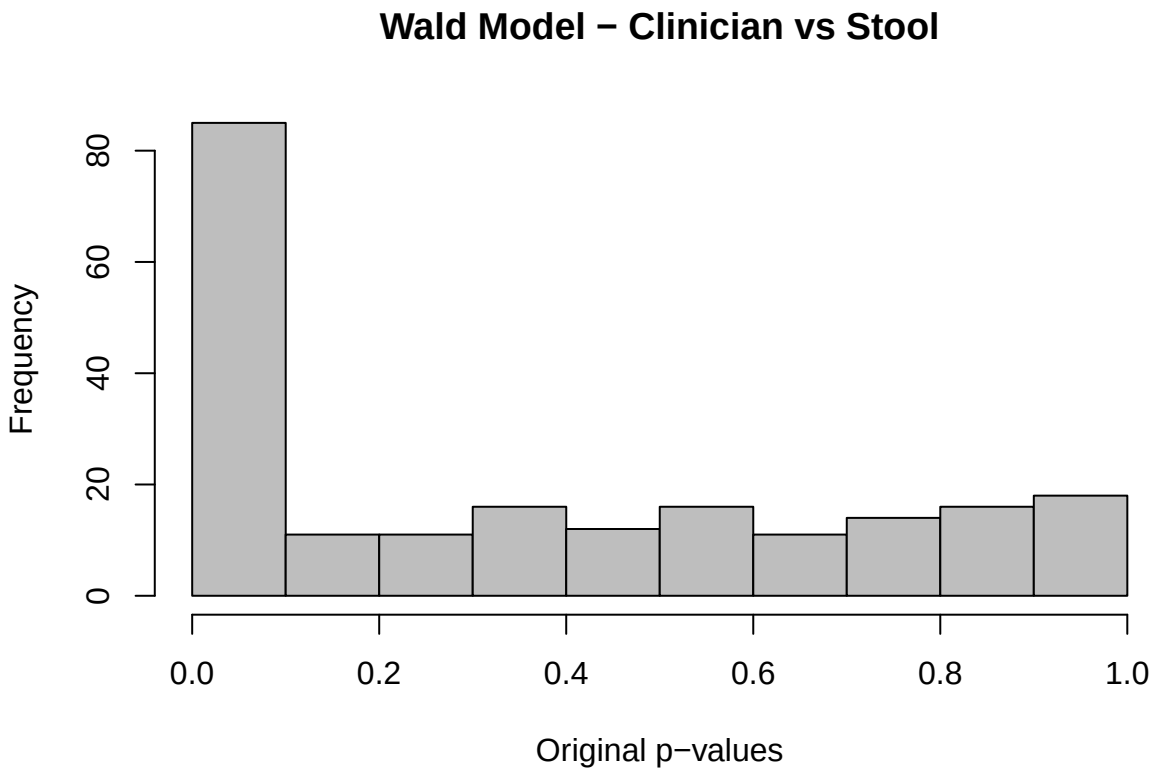

```r
resCTST_sig <- resCTST[(resCTST$padj < alpha), ]
resCTST_sig <- cbind(as(resCTST_sig, "data.frame"), as(tax_table(ps)[rownames(resCTST_sig), ], "matrix")
head(resCTST_sig)
```

```
##          baseMean log2FoldChange     lfcSE     stat       pvalue         padj
## ASV1068  8.191810       6.205783 0.9102231 6.817871 9.239974e-12 2.808952e-09
## ASV930  10.246362       7.072852 1.2769844 5.538714 3.046999e-08 4.631439e-06
## ASV473  55.398735       4.339785 0.7977984 5.439702 5.336990e-08 5.408149e-06
## ASV1129  8.010744       6.131689 1.1953911 5.129441 2.906035e-07 2.208587e-05
## ASV658  26.218338       4.915912 1.1726628 4.192093 2.763921e-05 1.473097e-03
## ASV1164 14.753210       6.139537 1.4685785 4.180598 2.907429e-05 1.473097e-03
##          Kingdom         Phylum               Class            Order
## ASV1068 Bacteria Proteobacteria Gammaproteobacteria Enterobacterales
## ASV930  Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV473  Bacteria Proteobacteria Gammaproteobacteria  Pseudomonadales
## ASV1129 Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV658  Bacteria Proteobacteria Gammaproteobacteria    Aeromonadales
## ASV1164 Bacteria Proteobacteria Gammaproteobacteria Enterobacterales
##                  Family                        Genus
## ASV1068     Yersiniaceae                      Yersinia
## ASV930    Clostridiaceae  Clostridium_sensu_stricto_5
## ASV473  Pseudomonadaceae                  Pseudomonas
## ASV1129   Clostridiaceae Clostridium_sensu_stricto_13
## ASV658    Aeromonadaceae                   Aeromonas
## ASV1164       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")
```

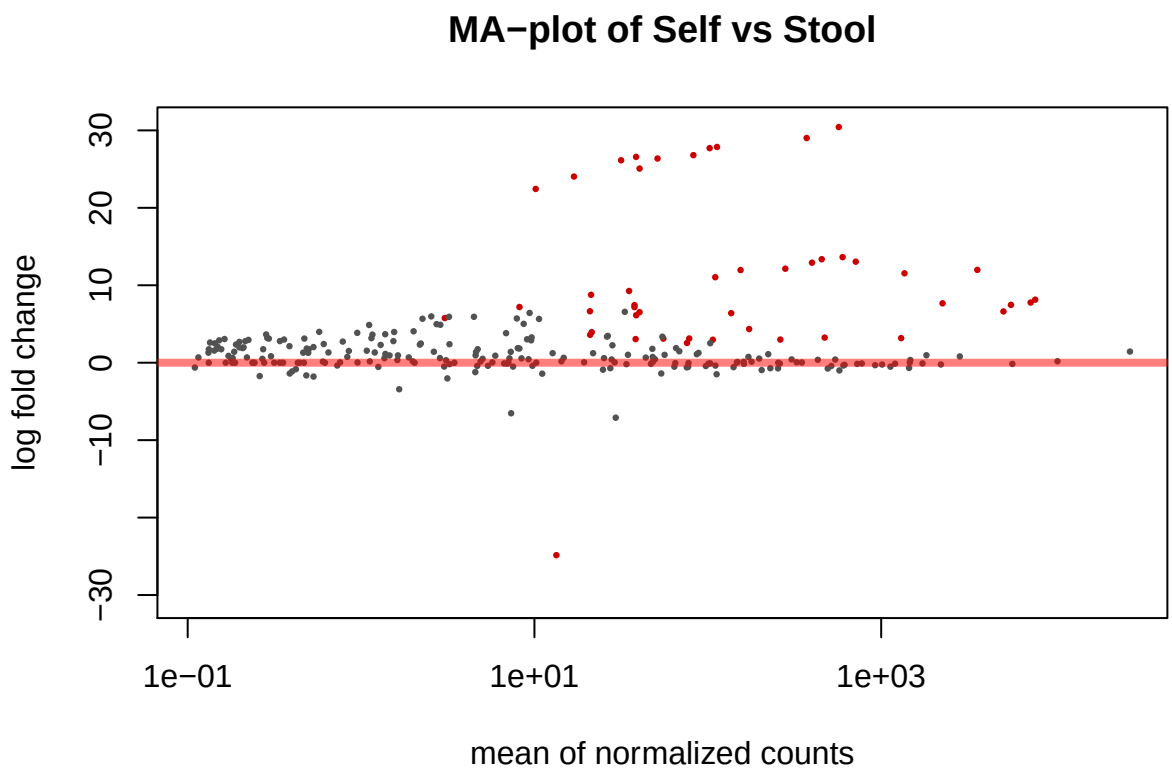



```
resCTS_sig <- resCTS[(resCTS$padj < alpha), ]
resCTS_sig <- cbind(as(resCTS_sig, "data.frame"), as(tax_table(ps)[rownames(resCTS_sig), ], "matrix"))
head(resCTS_sig)
```

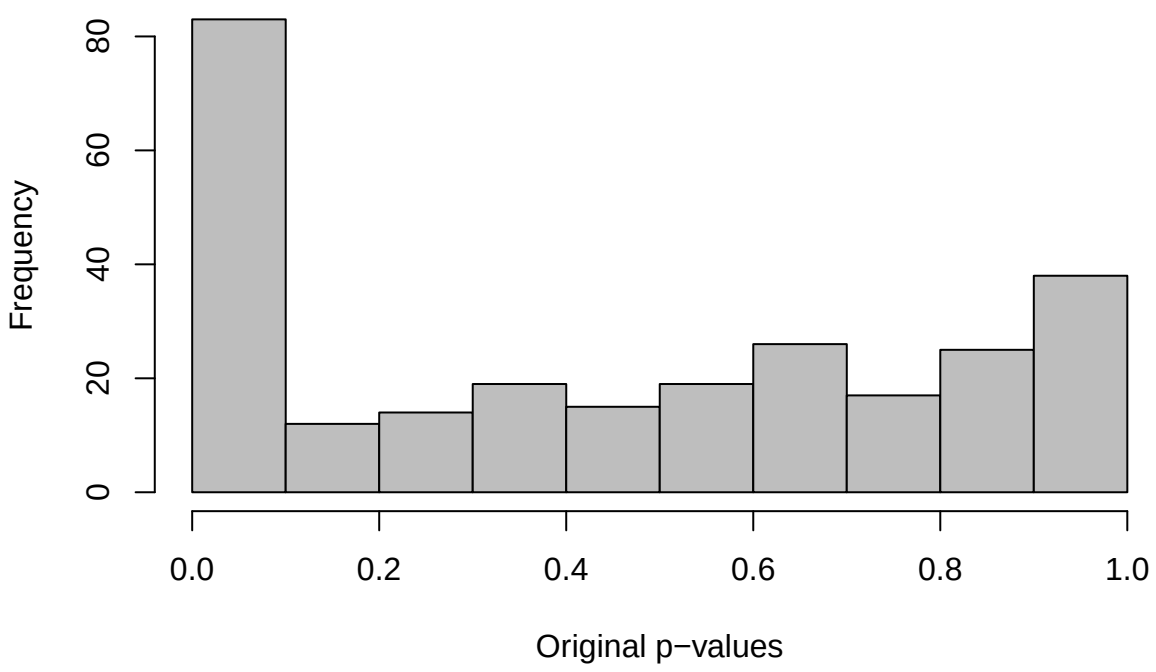
```
##          baseMean log2FoldChange     lfcSE     stat        pvalue          padj
## ASV324  371.68353       26.83913 1.1950890 22.45785 1.072464e-111 2.252174e-109
## ASV262  569.52608       28.29018 1.3237180 21.37176 2.447353e-101  2.569720e-99
## ASV365  102.45995       26.62138 1.5030199 17.71193  3.392565e-70  2.374796e-68
## ASV662  112.98959       26.83745 1.6813612 15.96174  2.360599e-57  1.239315e-55
## ASV283   82.54958       27.21862 1.7630615 15.43827  9.049272e-54  3.800694e-52
## ASV5   3585.31720       11.79910 0.7864814 15.00239  7.082408e-51  2.478843e-49
##         Kingdom           Phylum           Class
## ASV324 Bacteria       Firmicutes   Negativicutes
## ASV262 Bacteria       Firmicutes      Clostridia
## ASV365 Bacteria       Firmicutes      Clostridia
## ASV662 Bacteria       Firmicutes         Bacilli
## ASV283 Bacteria     Synergistota     Synergistia
## ASV5   Bacteria Campilobacterota Campylobacteria
##                                  Order
## ASV324      Veillonellales-Selenomonadales
## ASV262                      Clostridia_or
## ASV365 Peptostreptococcales-Tissierellales
## ASV662                     Lactobacillales
## ASV283                       Synergistales
## ASV5                     Campylobacterales
##                                    Family          Genus
## ASV324                     Veillonellaceae Negativicoccus
```

```
## ASV262                 Hungateiclostridiaceae Fastidiosipila
## ASV365 Peptostreptococcales-Tissierellales_fa      Gallicola
## ASV662                           Aerococcaceae      Facklamia
## ASV283                         Synergistaceae Pyramidobacter
## ASV5                       Campylobacteraceae  Campylobacter
```

```r
# 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")
```




```r
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)
```

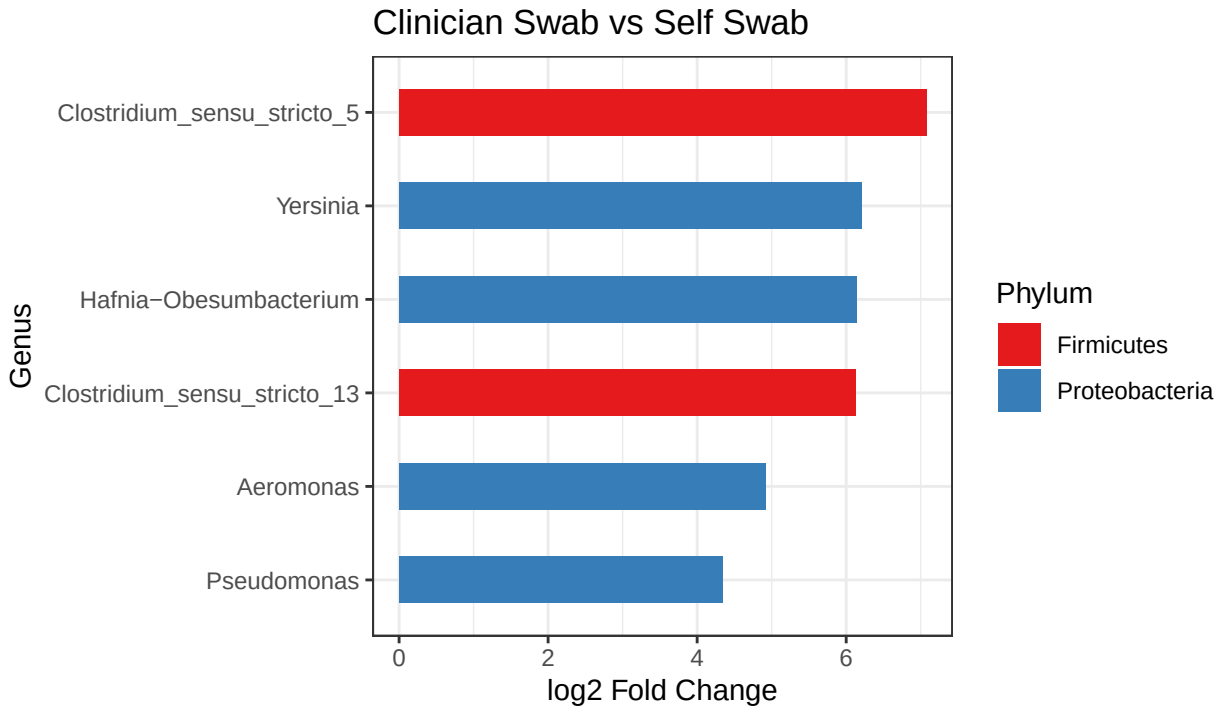
```
##          baseMean log2FoldChange     lfcSE     stat        pvalue          padj
## ASV324  371.68353       29.01258 1.1939331 24.30001 1.960067e-130 5.252978e-128
## ASV262  569.52608       30.42730 1.3230276 22.99823 4.854894e-117 6.505558e-115
## ASV365  102.45995       27.70469 1.5015082 18.45124  5.095997e-76  4.552424e-74
## ASV662  112.98959       27.85901 1.6801511 16.58125  9.522523e-62  6.380090e-60
## ASV5   3585.31720       11.98355 0.7864397 15.23772  1.986922e-52  1.064990e-50
## ASV283   82.54958       26.79915 1.7628105 15.20251  3.403086e-52  1.520045e-50
##         Kingdom           Phylum           Class
## ASV324 Bacteria       Firmicutes   Negativicutes
## ASV262 Bacteria       Firmicutes      Clostridia
## ASV365 Bacteria       Firmicutes      Clostridia
## ASV662 Bacteria       Firmicutes         Bacilli
## ASV5   Bacteria Campilobacterota Campylobacteria
## ASV283 Bacteria     Synergistota     Synergistia
##                                     Order
## ASV324      Veillonellales-Selenomonadales
## ASV262                        Clostridia_or
## ASV365 Peptostreptococcales-Tissierellales
## ASV662                      Lactobacillales
## ASV5                      Campylobacterales
## ASV283                        Synergistales
##                                       Family          Genus
## ASV324                        Veillonellaceae Negativicoccus
## ASV262                 Hungateiclostridiaceae Fastidiosipila
## ASV365 Peptostreptococcales-Tissierellales_fa      Gallicola
## ASV662                           Aerococcaceae      Facklamia
```

```
## ASV5                        Campylobacteraceae  Campylobacter
## ASV283                          Synergistaceae Pyramidobacter
```

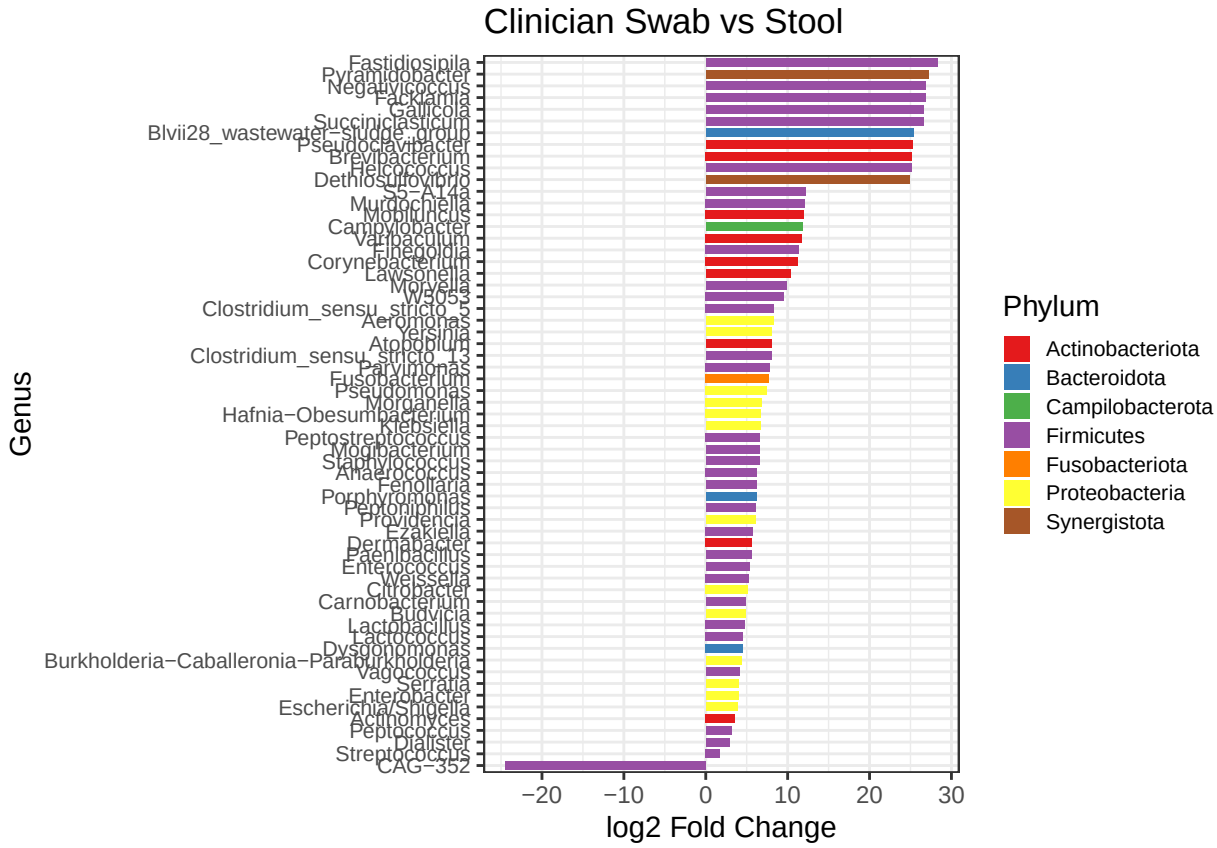
```
# Save .csv of significant fold change results
resCTST_sig$Comparison <- "Clinician Taken Swab vs Self Taken Swab"
resCTS_sig$Comparison <- "Clinician Taken Swab vs Stool"
resSTS_sig$Comparison <- "Self Taken Swab vs Stool"

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

Differential Abundance Figure

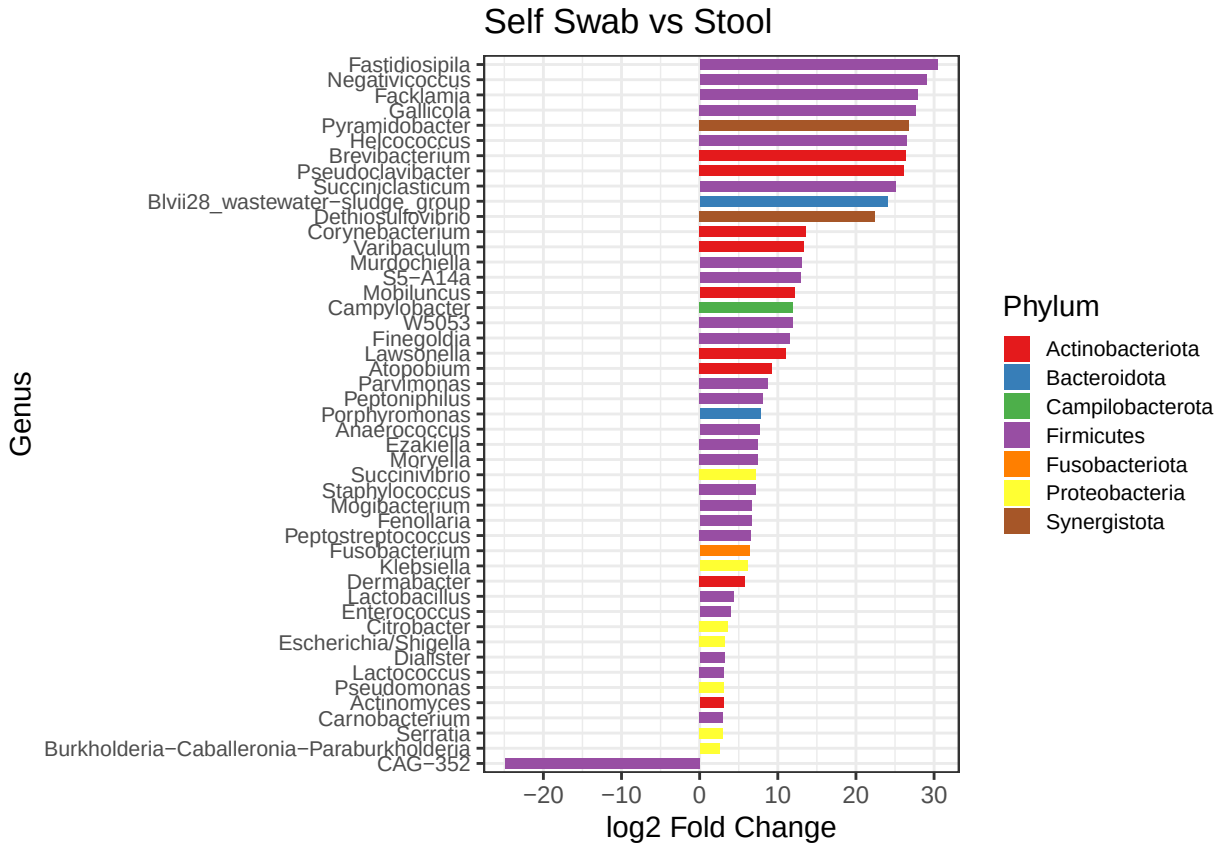
```
ggplot(resCTST_sig, aes(x = log2FoldChange, y = reorder(Genus, log2FoldChange), fill= Phylum)) +
  geom_bar(stat = "identity", position = "identity", width = 0.5) +
  labs(title = "Clinician Swab vs Self Swab", y = "Genus", x = "log2 Fold Change") +
  theme(aspect.ratio = 1) +
  scale_fill_brewer(palette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



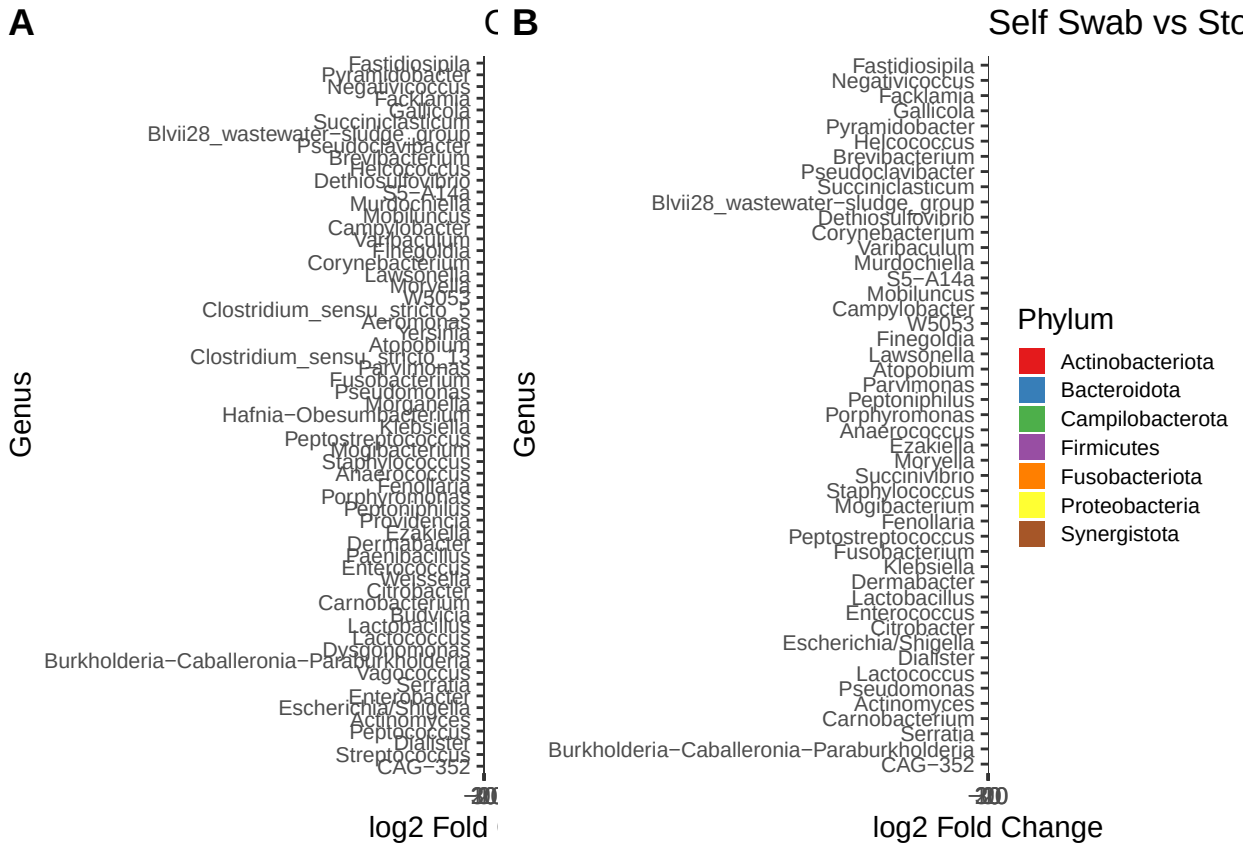

```
ggsave("Purcell Final/Final_results/5)Differential_Abundance_clinVSself.pdf", width = 7, height = 4)

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 = "Set1") +
  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 = "Set1") +
  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")
```



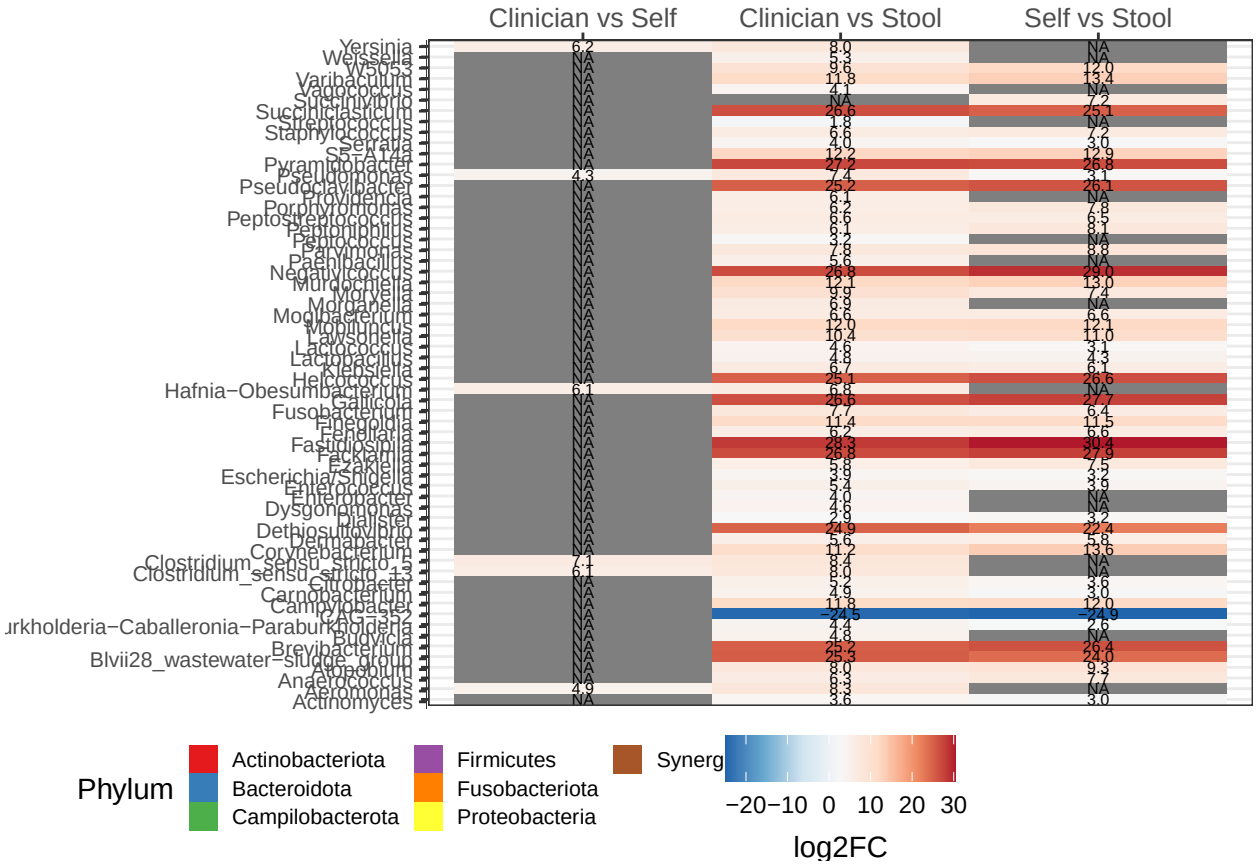

```
ggsave("Purcell Final/Final_results/5)Differential_Abundance_swabsVSstool.pdf", width = 12, height = 8)
```

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 <- t(heat)
heat <- as.data.frame(heat)
colnames(heat) <- c("CTS", "CTS_phylum", "CTS_genus",
                    "STS", "STS_phylum", "STS_genus",
                    "CTST", "CTST_phylum", "CTST_genus")
```

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

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

heat <- select(heat, -CTS_genus, -STS_genus, -CTST_genus, -CTS_phylum, -STS_phylum, -CTST_phylum)

# file for ggplot based heatmap
SamplingComparison <- c(1:(nrow(heat)*3))
SamplingComparison[1:nrow(heat)] <- "CTS"
SamplingComparison[(nrow(heat)+1):(nrow(heat)*2)] <- "STS"
SamplingComparison[((nrow(heat)*2)+1):(nrow(heat)*3)] <- "CTST"
log2FC <- c(1:(nrow(heat)*3))
log2FC[1:nrow(heat)] <- as.numeric(as.character(heat$CTS))
log2FC[(nrow(heat)+1):(nrow(heat)*2)] <- as.numeric(as.character(heat$STS))
log2FC[((nrow(heat)*2)+1):(nrow(heat)*3)] <- as.numeric(as.character(heat$CTST))
Phylum <- c(1:(nrow(heat)*3))
Phylum[1:nrow(heat)] <- heat$sigPhylum
Phylum[(nrow(heat)+1):(nrow(heat)*2)] <- heat$sigPhylum
Phylum[((nrow(heat)*2)+1):(nrow(heat)*3)] <- heat$sigPhylum
Genus <- c(1:(nrow(heat)*3))
Genus[1:nrow(heat)] <- heat$sigGenus
Genus[(nrow(heat)+1):(nrow(heat)*2)] <- heat$sigGenus
Genus[((nrow(heat)*2)+1):(nrow(heat)*3)] <- heat$sigGenus
ftp <- as.data.frame(cbind(SamplingComparison, log2FC, Phylum, Genus))

ftp$log2FC <- as.numeric(as.character(ftp$log2FC))
ftp$SamplingComparison <- factor(ftp$SamplingComparison, levels = c("CTST", "CTS", "STS"))

heatLog <- ggplot(ftp, aes(SamplingComparison, Genus, fill = log2FC)) + geom_tile() +
  geom_text(aes(label = sprintf("%2.1f", log2FC)), size = 2) +
  theme(axis.title = element_blank(), legend.position = "bottom",
        axis.text.y = element_blank(),
        axis.text.x = element_text(family = "Helvetica", size = 10, face = "plain"),
        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_x_discrete(position = "top", labels = (c("Clinician vs Self",
                                                 "Clinician vs Stool",
                                                 "Self vs Stool")))

heatPhylum <- ggplot(ftp, aes(SamplingComparison, Genus, fill = Phylum)) + geom_tile() +
  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 = "Set1", guide = guide_legend(ncol = 3))
```

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




```
ggsave("Purcell Final/Final_results/6)Differential_Abundance_heatmap.pdf", width = 8, height = 8)
```

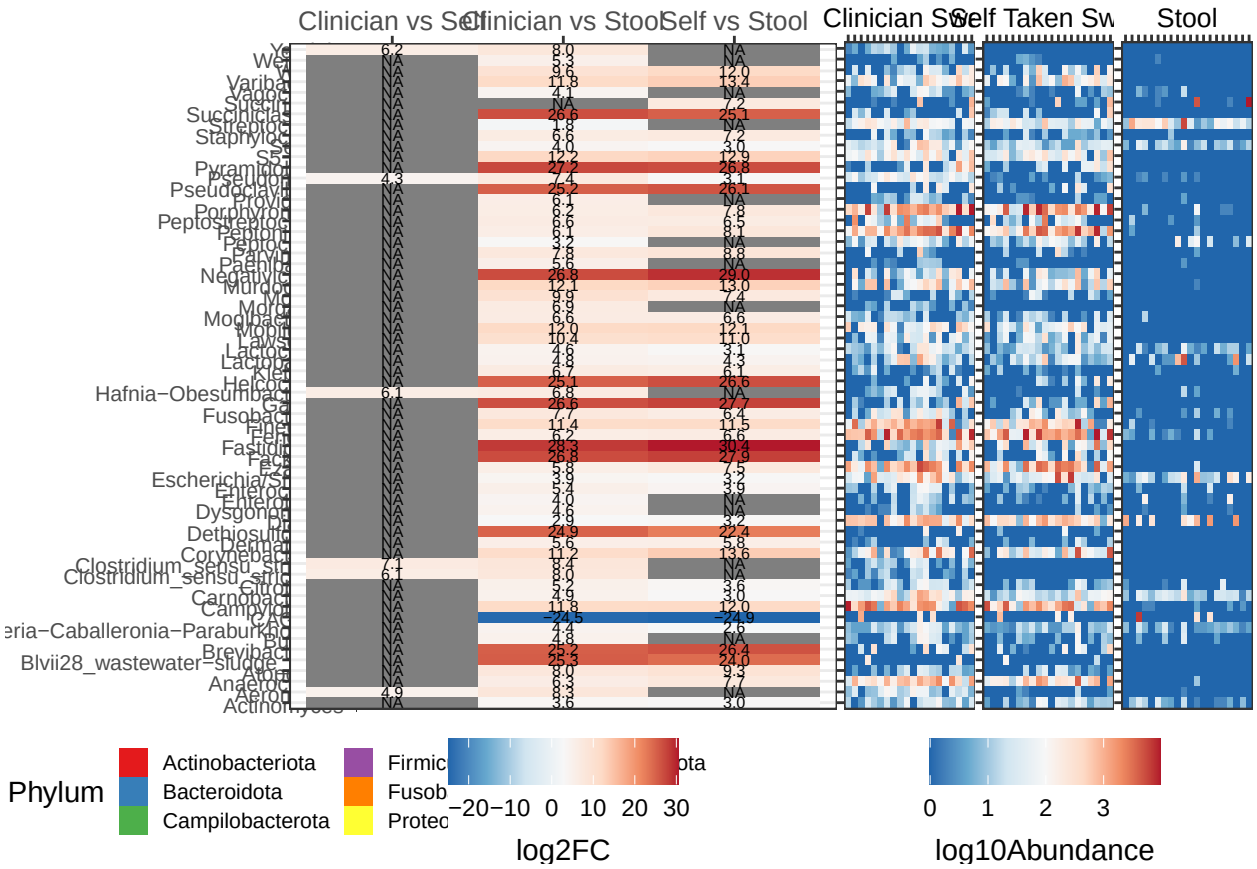
Species Abundance ggplot Heatmap

```
# Make figure with individual abundance to go next to heatmap
heat_ps <- subset_taxa(ps_rare, Genus %in% heat$sigGenus)
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") + xlab("Clinician Swab") +
  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 = 0, b = 0, l = 0, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_distiller(palette = "RdBu") +
  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") + xlab("Self Taken Swab") +
  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 = 0, b = 0, l = 0, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_distiller(palette = "RdBu") +
  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") + xlab("Stool") +
  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 = 0, b = 0, l = 0, unit = "pt"),
        legend.margin = margin(t = 11, r = 0, b = 0, l = 0, unit = "pt")) +
  scale_fill_distiller(palette = "RdBu") +
  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("Purcell Final/Final_results/7)Differential_Abundance_heatmap_extra.pdf",width = 11, height = 8)
```

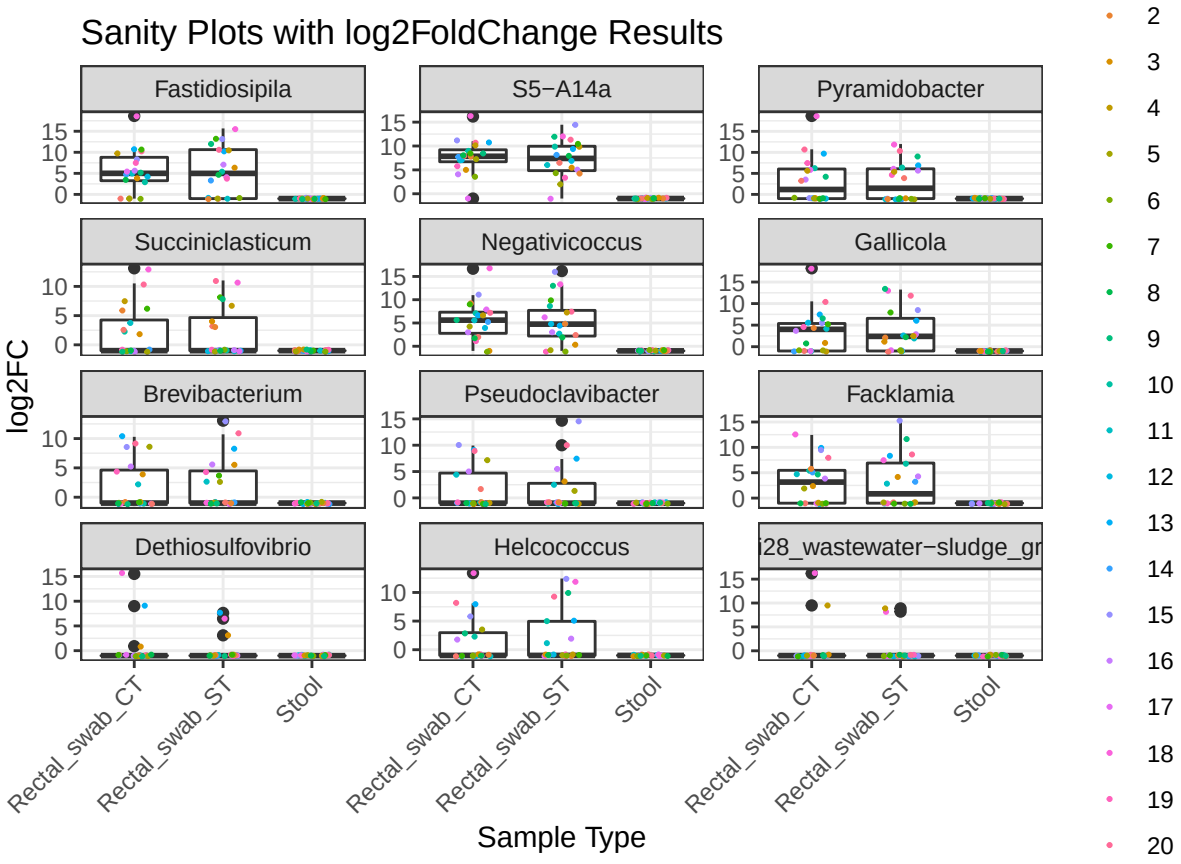
Boxplot Sanity Checks

```
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") %>%
  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	ASV262	3.017179
10B	Rectal_swab_ST	10	ASV262	5.359164
10C	Stool	10	ASV262	-1.000000
11A	Rectal_swab_CT	11	ASV262	4.888552
11B	Rectal_swab_ST	11	ASV262	-1.000000
11C	Stool	11	ASV262	-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.
```



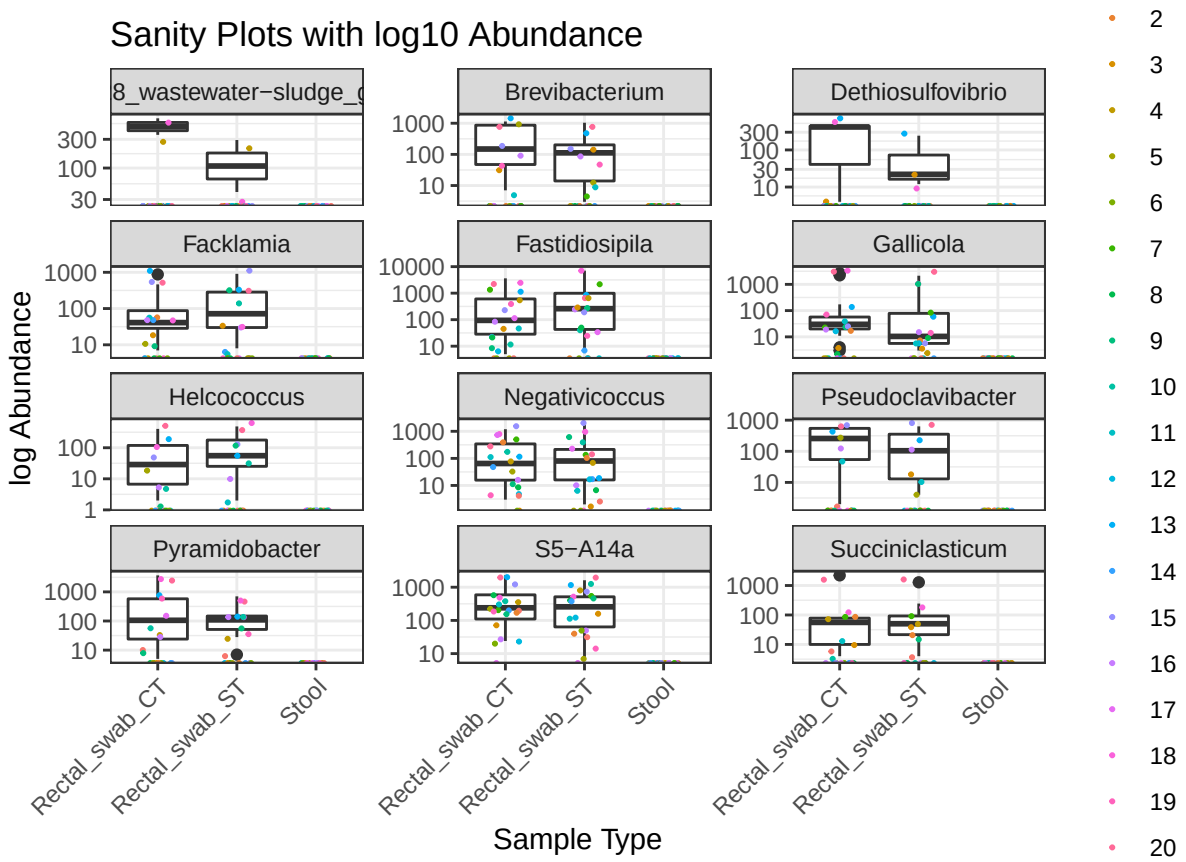

```
ggsave("Purcell Final/Final_results/8)Sanity_FoldChange_plots.pdf", width = 7, height = 8)

# 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("Purcell Final/Final_results/8)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 3.6.3 (2020-02-29)
```

```
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/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] DESeq2_1.26.0               SummarizedExperiment_1.16.1
##  [3] DelayedArray_0.12.3         BiocParallel_1.20.1
##  [5] matrixStats_0.56.0          Biobase_2.46.0
##  [7] GenomicRanges_1.38.0        GenomeInfoDb_1.22.1
##  [9] IRanges_2.20.2              S4Vectors_0.24.4
## [11] BiocGenerics_0.32.0         scales_1.1.1
## [13] tidyr_1.1.0                 ggpubr_0.4.0
## [15] extrafont_0.17              ggplot2_3.3.2
## [17] phyloseq_1.30.0             dplyr_1.0.0
## [19] vegan_2.5-6                 permute_0.9-5
## [21] Rmisc_1.5                   plyr_1.8.6
## [23] lattice_0.20-41             RColorBrewer_1.1-2
##
## loaded via a namespace (and not attached):
##   [1] colorspace_1.4-1       ggsignif_0.6.0         ellipsis_0.3.1
##   [4] rio_0.5.16             htmlTable_2.0.1        XVector_0.26.0
##   [7] base64enc_0.1-3        rstudioapi_0.11        farver_2.0.3
##  [10] bit64_0.9-7.1          AnnotationDbi_1.48.0   codetools_0.2-16
##  [13] splines_3.6.3          geneplotter_1.64.0     knitr_1.29
##  [16] ade4_1.7-15            Formula_1.2-3          jsonlite_1.7.0
##  [19] annotate_1.64.0        broom_0.7.0            Rttf2pt1_1.3.8
##  [22] cluster_2.1.0          png_0.1-7              compiler_3.6.3
##  [25] backports_1.1.8        Matrix_1.2-18          acepack_1.4.1
##  [28] htmltools_0.5.0        tools_3.6.3            igraph_1.2.5
##  [31] gtable_0.3.0           glue_1.4.1             GenomeInfoDbData_1.2.2
##  [34] reshape2_1.4.4         Rcpp_1.0.5             carData_3.0-4
##  [37] cellranger_1.1.0       vctrs_0.3.2            Biostrings_2.54.0
##  [40] multtest_2.42.0        ape_5.4                nlme_3.1-148
##  [43] extrafontdb_1.0        iterators_1.0.12       xfun_0.15
##  [46] stringr_1.4.0          openxlsx_4.1.5         lifecycle_0.2.0
##  [49] XML_3.99-0.3           rstatix_0.6.0          zlibbioc_1.32.0
##  [52] MASS_7.3-51.6          hms_0.5.3              biomformat_1.14.0
##  [55] rhdf5_2.30.1           yaml_2.2.1             curl_4.3
##  [58] memoise_1.1.0          gridExtra_2.3          rpart_4.1-15
##  [61] RSQLite_2.2.0          latticeExtra_0.6-29    stringi_1.4.6
##  [64] highr_0.8              genefilter_1.68.0      foreach_1.5.0
##  [67] checkmate_2.0.0        zip_2.0.4              rlang_0.4.7
##  [70] pkgconfig_2.0.3        bitops_1.0-6           evaluate_0.14
##  [73] purrr_0.3.4            Rhdf5lib_1.8.0         labeling_0.3
```

```
##  [76] htmlwidgets_1.5.1      cowplot_1.0.0          bit_1.1-15.2
##  [79] tidyselect_1.1.0       magrittr_1.5           R6_2.4.1
##  [82] generics_0.0.2         Hmisc_4.4-0            DBI_1.1.0
##  [85] pillar_1.4.6           haven_2.3.1            foreign_0.8-76
##  [88] withr_2.2.0            mgcv_1.8-31            survival_3.2-3
##  [91] abind_1.4-5            RCurl_1.98-1.2         nnet_7.3-14
##  [94] tibble_3.0.3           crayon_1.3.4           car_3.0-8
##  [97] rmarkdown_2.3          jpeg_0.1-8.1           locfit_1.5-9.4
## [100] grid_3.6.3             readxl_1.3.1           data.table_1.12.8
## [103] blob_1.2.1             forcats_0.5.0          digest_0.6.25
## [106] xtable_1.8-4           munsell_0.5.0
```