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 ]
```

Filter out Yersinia

```
ps <- subset_taxa(ps, Genus != "Yersinia")
```

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, 1:150 * 1000),
                                                           each = 10))
summary(rarefaction_curve_data)
saveRDS(rarefaction_curve_data, file = "Purcell Final/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)
```

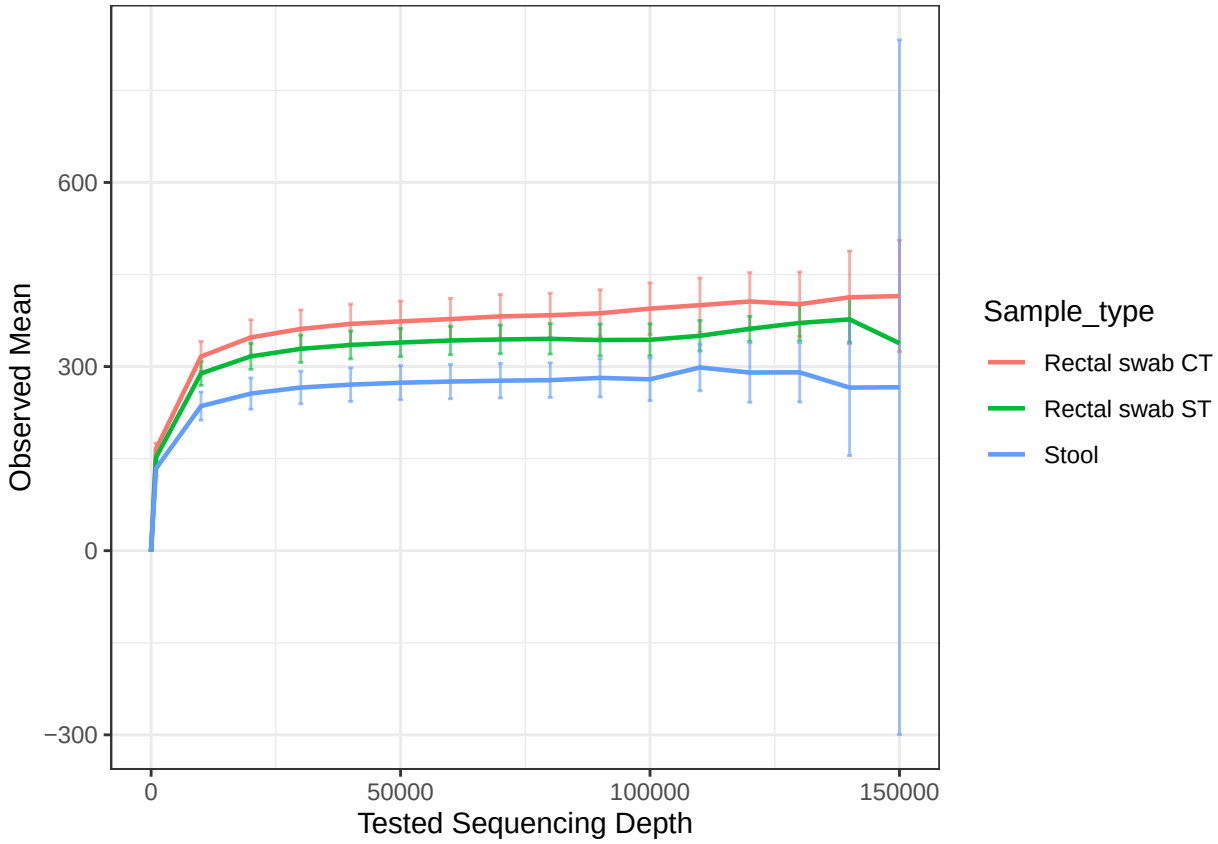
```
##      Depth            Sample          Measure      Alpha_diversity
##  Min.   :     1   X13A   : 1520   Observed:74070   Min.   :  1.0
##  1st Qu.: 29000   X14A   : 1520                    1st Qu.:263.0
##  Median : 60000   X14B   : 1520                    Median :331.0
##  Mean   : 62475   X15A   : 1520                    Mean   :320.4
##  3rd Qu.: 92000   X15C   : 1520                    3rd Qu.:378.0
##  Max.   :150000   X18A   : 1520                    Max.   :546.0
##                   (Other):64950
```

```
# 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"))
```

```
## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced
```

```
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

## ...

## 1300TUs 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
##  96045 149391 133488 101852  87122 135639  60955 141908 134481 164730 149672
##    13C    14A    14B    14C    15A    15B    15C    16A    16B    16C    17A
## 141290 154794 167659 131939 167015 106082 161420 111021 121601 134735 149622
##    17B    17C    18A    18B    18C    19A    19B    19C     1A     1B     1C
## 123675  91485 154502 123487 153504 152567 146504 115610 136572  75035  89197
##    20A    20B    20C     2A     2B     2C     3A     3B     3C     4A     4B
## 187470 110105 106713 107518 114008 100238 136162 126355 100973  98108 118848
##     4C     5A     5B     5C     6A     6B     6C     7A     7B     7C     8A
## 113301 153840 113210  84418 127131 147421  88442 134264 136792 114586 127880
##     8B     8C     9A     9B     9C
##  88579  99543  82179 139662 105460
```

```
sample_sums(ps_rare)
```

```
##   10A   10B   10C   11A   11B   11C   12A   12B   12C   13A   13B   13C   14A
## 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859
##   14B   14C   15A   15B   15C   16A   16B   16C   17A   17B   17C   18A   18B
## 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859
##   18C   19A   19B   19C    1A    1B    1C   20A   20B   20C    2A    2B    2C
## 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859
##    3A    3B    3C    4A    4B    4C    5A    5B    5C    6A    6B    6C    7A
## 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859 54859
##    7B    7C    8A    8B    8C    9A    9B    9C
## 54859 54859 54859 54859 54859 54859 54859 54859
```

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 = 0.98663, p-value = 0.7541
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  alpha_summary$Shannon
## W = 0.97173, p-value = 0.1773
```

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

f <- c("Clinician", "Self", "Stool") # treatment levels
```

```r
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 108335   54168  21.996 4.51e-07 ***
## blk         19 120686    6352   2.579  0.00635 **
## Residuals   38  93578    2463
## ---
## 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  0.994  0.4970   8.835 0.000706 ***
## blk         19  4.005  0.2108   3.747 0.000260 ***
## Residuals   38  2.137  0.0562
## ---
## 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 0.0759         -
## Stool          6.3e-06        0.0015
##
## P value adjustment method: BH
```

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

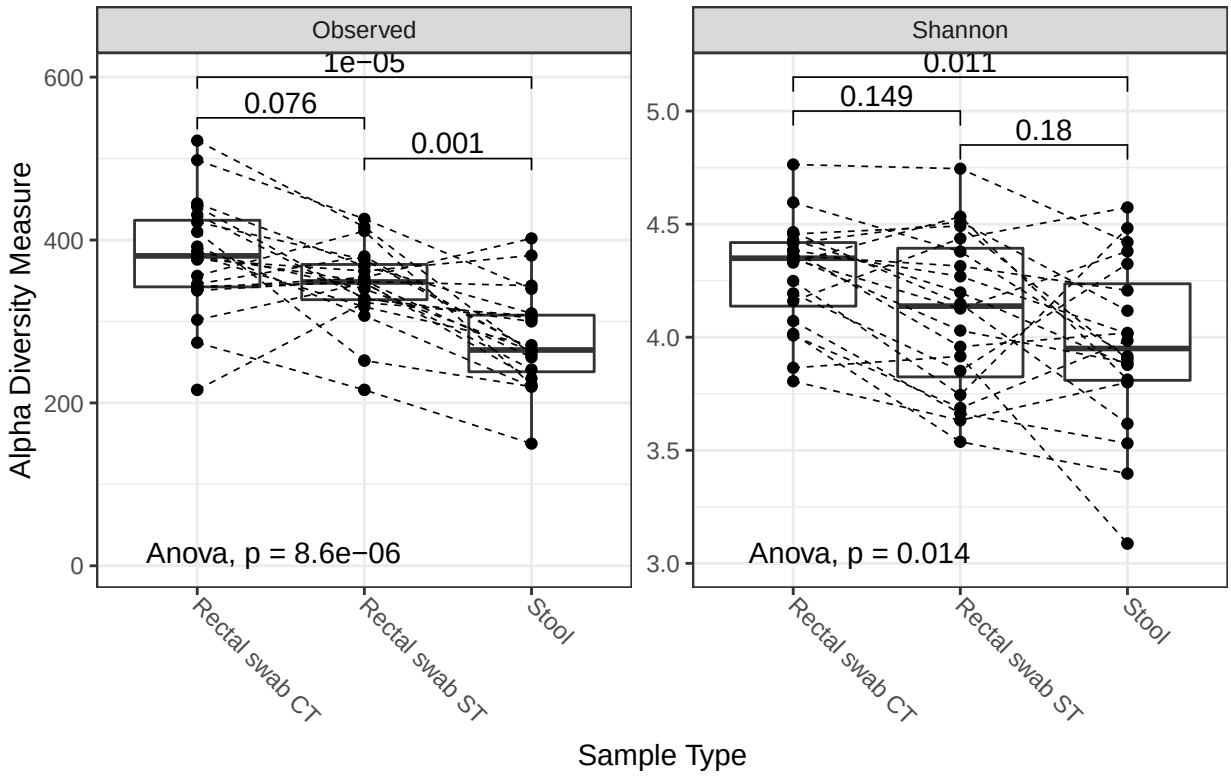
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  alpha_summary$Shannon and sample_data(ps_rare)$Sample_type
##
##                Rectal swab CT Rectal swab ST
## Rectal swab ST 0.149          -
## Stool          0.011          0.180
##
## P value adjustment method: BH
```

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

```r
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$p.value[2,1], 5), round(pObs$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(600, 500, 550, 5.15, 4.85, 5.0)

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

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

```r
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)
```




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

Beta Diversity - Bray-Curtis

```
# 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")
```

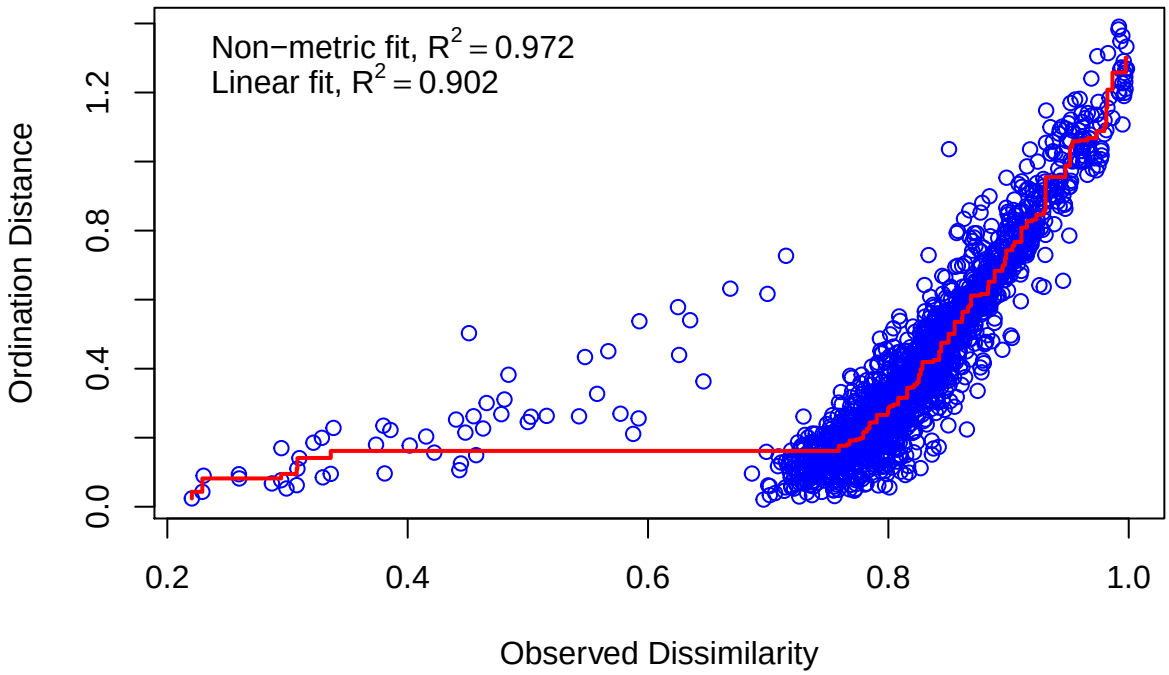
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1660649
## Run 1 stress 0.1742839
## Run 2 stress 0.1690271
## Run 3 stress 0.1963773
## Run 4 stress 0.1689969
## Run 5 stress 0.1660679
## ... Procrustes: rmse 0.0007082043  max resid 0.00380969
## ... Similar to previous best
## Run 6 stress 0.179229
## Run 7 stress 0.1805837
## Run 8 stress 0.1690662
## Run 9 stress 0.1909203
## Run 10 stress 0.1688625
## Run 11 stress 0.1660649
## ... New best solution
## ... Procrustes: rmse 0.0001955691  max resid 0.001334372
## ... Similar to previous best
```

```
## Run 12 stress 0.1660682
## ... Procrustes: rmse 0.0007703815  max resid 0.004055105
## ... Similar to previous best
## Run 13 stress 0.1660648
## ... New best solution
## ... Procrustes: rmse 0.0001232208  max resid 0.000492369
## ... Similar to previous best
## Run 14 stress 0.1660648
## ... New best solution
## ... Procrustes: rmse 5.110814e-05  max resid 0.0002284743
## ... Similar to previous best
## Run 15 stress 0.1689939
## Run 16 stress 0.1689967
## Run 17 stress 0.1744556
## Run 18 stress 0.1744547
## Run 19 stress 0.1976383
## Run 20 stress 0.1688245
## *** 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:     0.1660648
## 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    1.4193 0.70965  2.1981 0.0716  0.002 **
## Residuals                        57   18.4025 0.32285         0.9284
## Total                            59   19.8218                 1.0000
## ---
## 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   14.6257 0.76977  5.9258 0.73786  0.001 ***
```

```
## Residuals                       40    5.1961 0.12990         0.26214
## Total                           59   19.8218                 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)$Sample_type                                  2    1.4193
## sample_data(ps_rare)$Individual                                  19   14.6257
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 38    3.7768
## Residuals                                                         0    0.0000
## Total                                                            59   19.8218
##                                                                  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)$Sample_type                                 0.07160      1
## sample_data(ps_rare)$Individual                                  0.73786      1
## sample_data(ps_rare)$Sample_type:sample_data(ps_rare)$Individual 0.19054      1
## Residuals                                                        0.00000
## Total                                                            1.00000
```

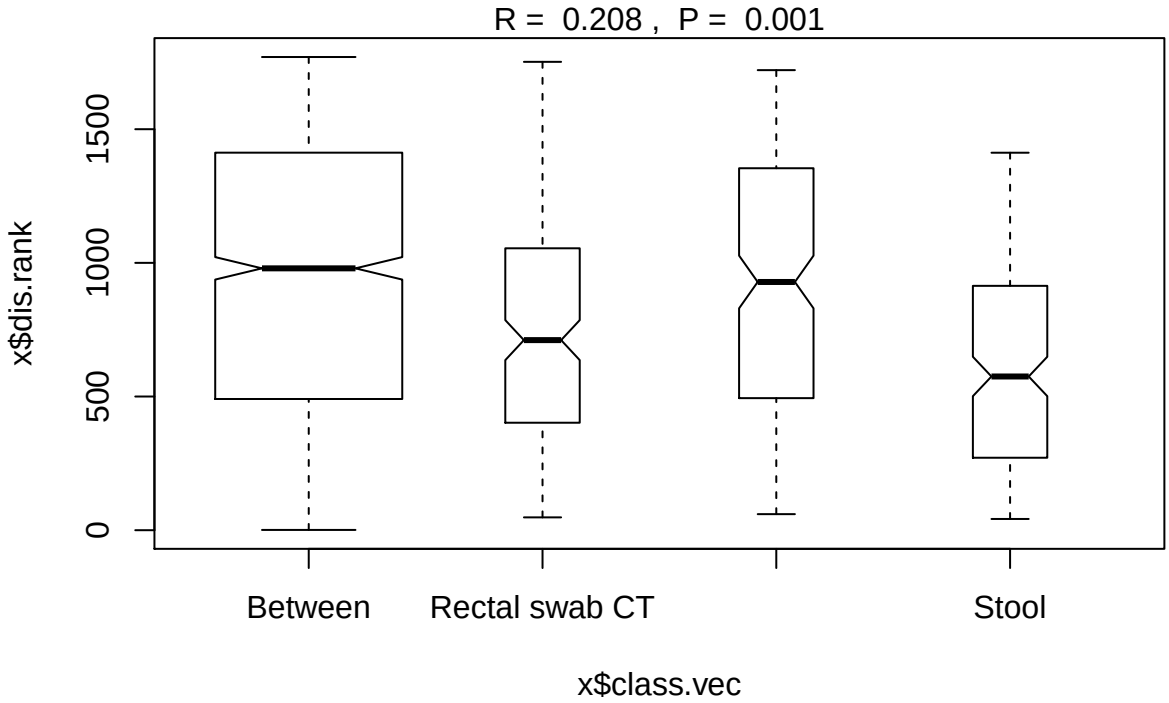
```
anosim(bray_dist, sample_data(ps_rare)$Sample_type)
```

```
##
## Call:
## anosim(x = bray_dist, grouping = sample_data(ps_rare)$Sample_type)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.2083
##       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: 0.2083
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0316 0.0487 0.0642 0.0846
##
## Dissimilarity ranks between and within classes:
##                0%     25%   50%      75% 100%    N
## Between         1 491.375 979.5 1411.625 1770 1200
## Rectal swab CT 48 402.500 711.0 1053.125 1752  190
## Rectal swab ST 60 495.250 928.5 1352.125 1721  190
## Stool          42 271.500 575.0  913.000 1412  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: 0.7739
##       Significance: 0.001
```

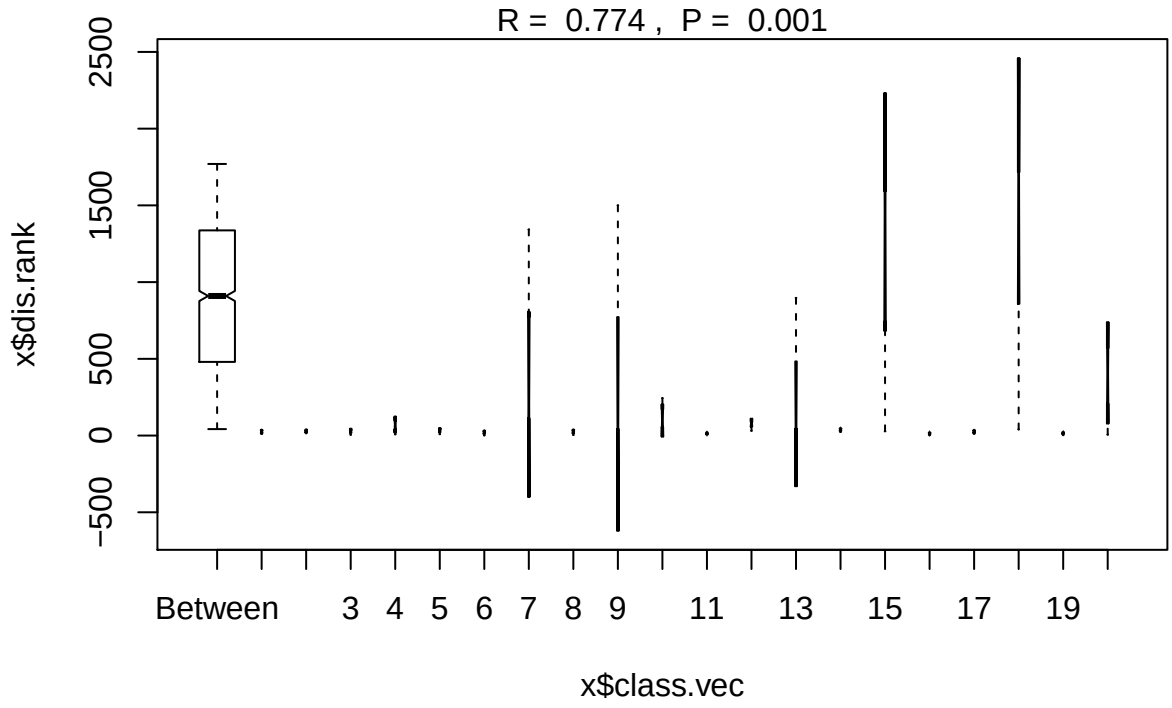
```
##
## Permutation: free
## Number of permutations: 999
```

```
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: 0.7739
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0855 0.1118 0.1381 0.1564
##
## Dissimilarity ranks between and within classes:
##         0%    25%    50%     75%   100%    N
## Between 42 480.25  909.5 1336.75 1770.0 1710
## 1       10  18.00   26.0   30.50   35.0    3
## 2       15  22.00   29.0   33.00   37.0    3
## 3        5  18.50   32.0   32.50   33.0    3
## 4        8  39.50   71.0   97.00  123.0    3
## 5        9  22.50   36.0   37.00   38.0    3
## 6        1  11.50   22.0   23.00   24.0    3
## 7       19 111.00  203.0  773.00 1343.0    3
## 8        4  15.50   27.0   28.50   30.0    3
## 9       41  42.50   44.0  772.00 1500.0    3
## 10      11  54.50   98.0  171.00  244.0    3
## 11       7  10.00   13.0   17.00   21.0    3
## 12      31  58.00   85.0   86.50   88.0    3
## 13      20  44.50   69.0  483.00  897.0    3
## 14      23  31.00   39.0   41.00   43.0    3
## 15      28 742.50 1457.0 1592.75 1728.5    3
## 16       2   8.00   14.0   15.50   17.0    3
## 17      12  18.50   25.0   29.50   34.0    3
## 18      40 858.50 1677.0 1717.25 1757.5    3
## 19       3   9.50   16.0   17.00   18.0    3
## 20       6 207.25  408.5  570.25  732.0    3
```

```
plot(anoInd)
```

```
## Warning in bxp(list(stats = structure(c(42, 480, 909.5, 1337, 1770, 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 0.01114 0.0055698  2.7521 0.07228 .
## Residuals 57 0.11536 0.0020238
## ---
## 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 0.01114 0.0055698 2.7521    999  0.072 .
## Residuals 57 0.11536 0.0020238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

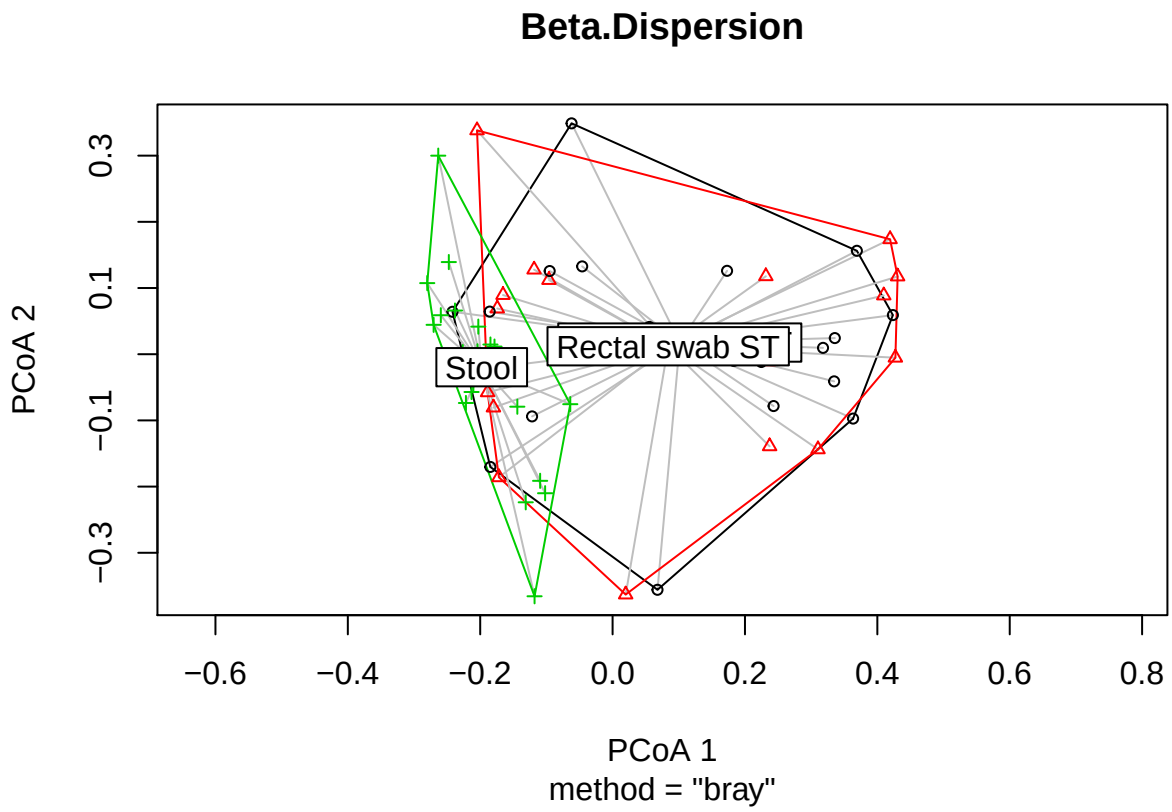
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
```

```
## Response: Distances
##           Df  Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     2 0.01114 0.0055698 2.7521    999  0.078 .
## Residuals 57 0.11536 0.0020238
## ---
## 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                       0.161000 0.352
## Rectal swab ST       0.163939                0.030
## Stool                0.352525        0.027001
```

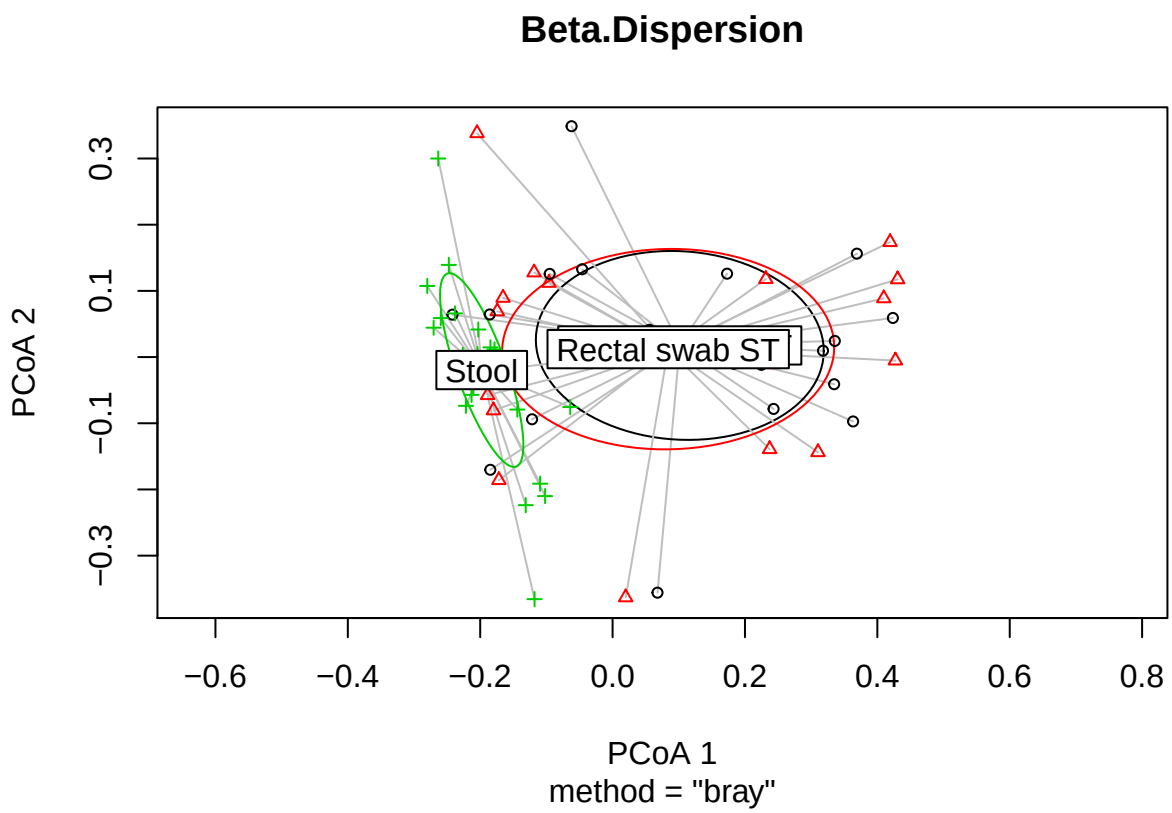
```
TukeyHSD(ps.disper)
```

```
##   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  0.01962352 -0.01461033 0.053857382 0.3583522
## Stool-Rectal swab CT          -0.01356902 -0.04780288 0.020664837 0.6088017
## Stool-Rectal swab ST          -0.03319255 -0.06742640 0.001041313 0.0592313
```

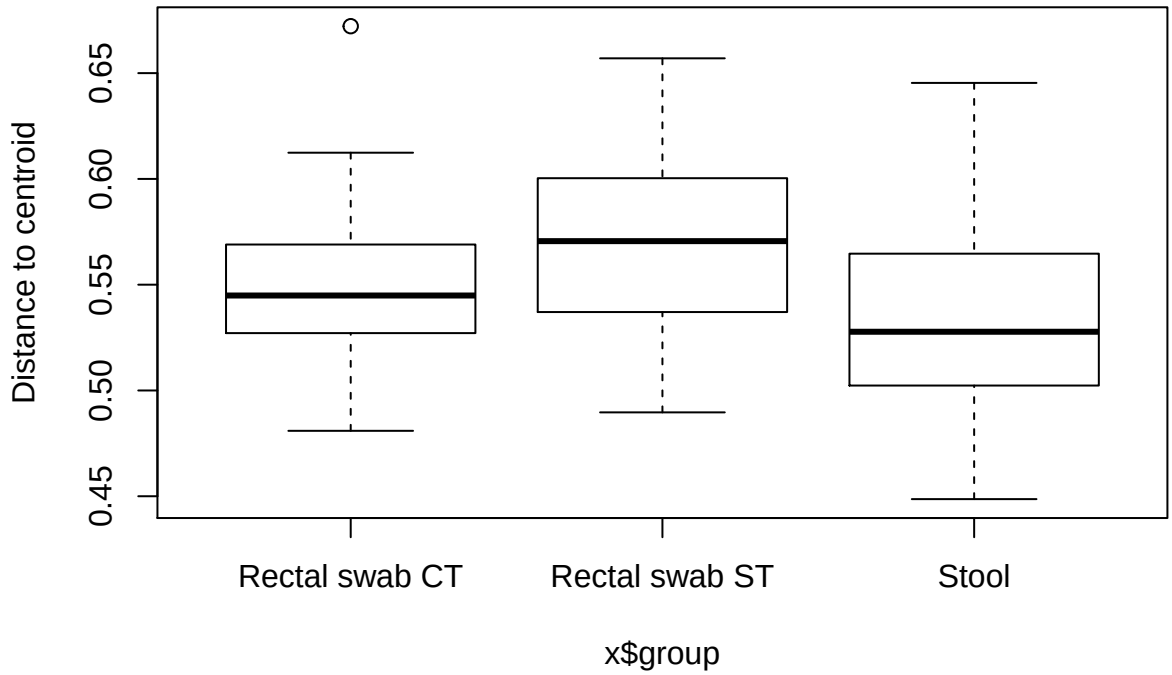
```
# Beta Dispersion Plots
Beta.Dispersion <- ps.disper
plot(Beta.Dispersion)
```



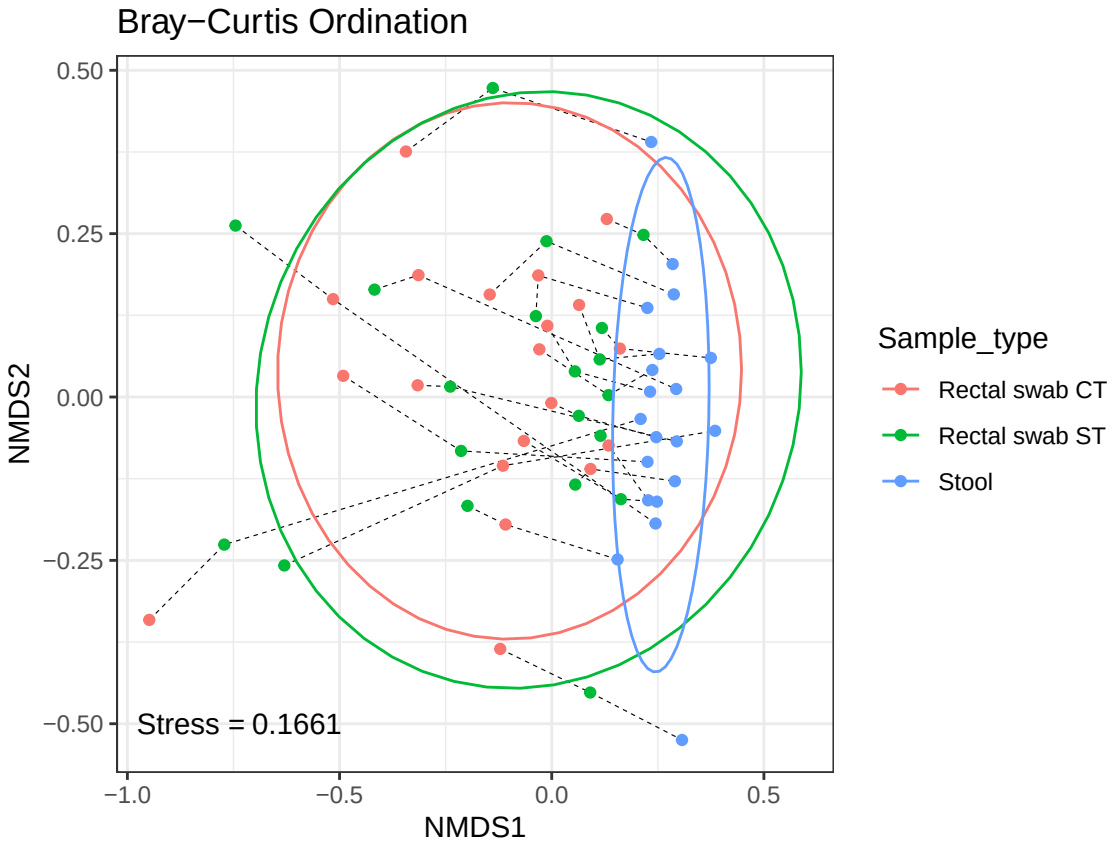

```
plot(Beta.Dispersion, hull = FALSE, ellipse = TRUE)
```



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




```r
# 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 = -0.85, y = -0.5, label = "Stress =") +
  annotate("text", x = -0.6, y = -0.5, label = round(ord.nmds.bray$stress, 4)) +
  stat_ellipse(aes(color = Sample_type)) +
  ggtitle("Bray-Curtis Ordination") +
  theme(aspect.ratio = 1)
```



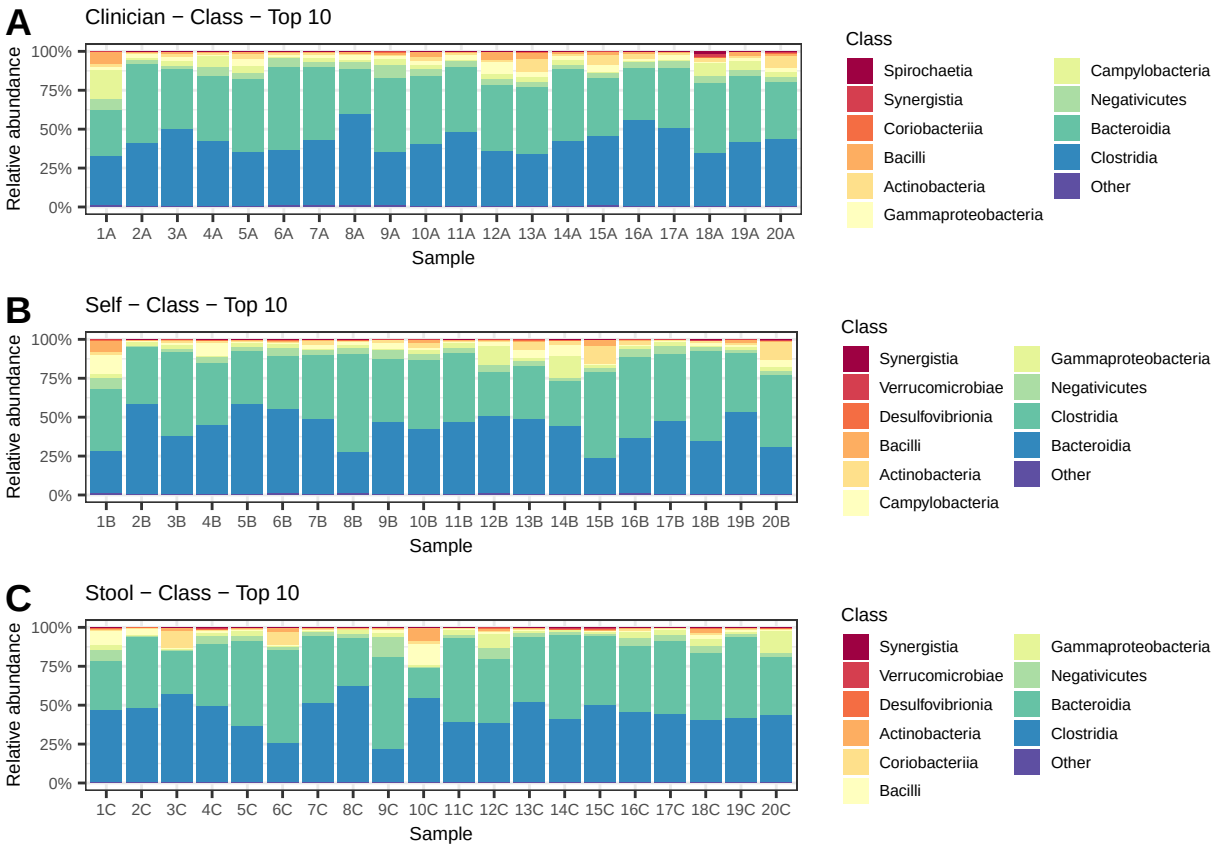

```
ggsave("Purcell Final/Final_results/3)Beta_Diversity.pdf", width = 6, height = 4.5)
```

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"

# 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"))
```

```r
# 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
```

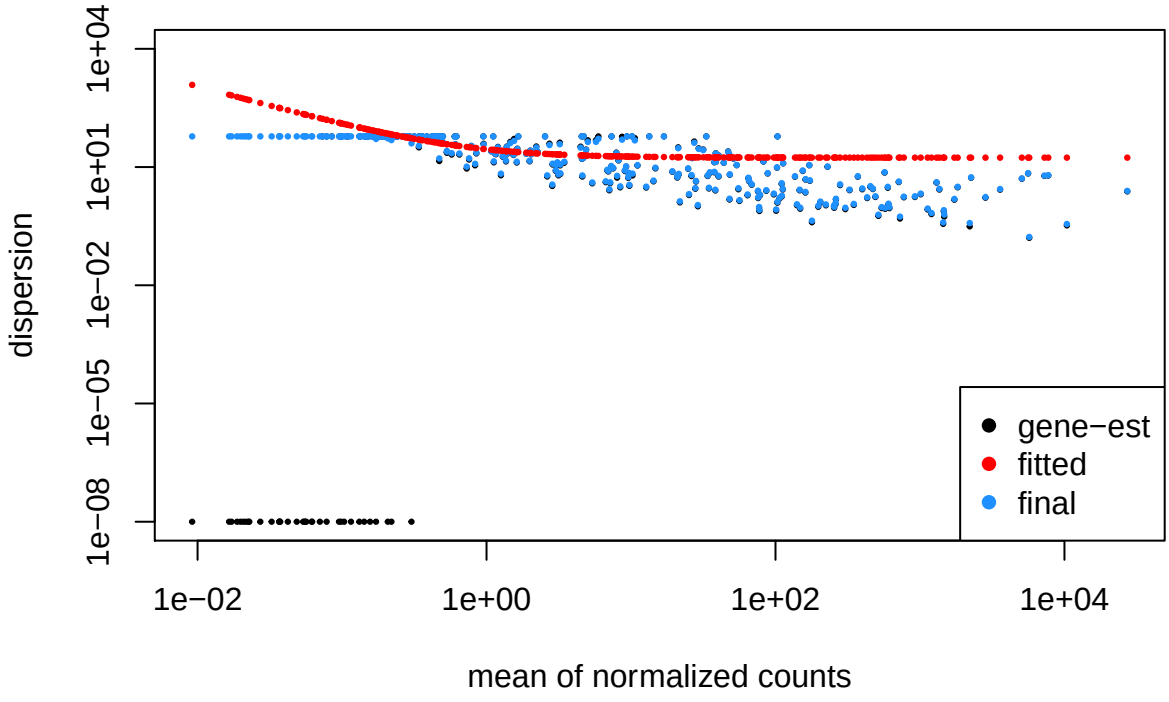
```
## fitting model and testing
```

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

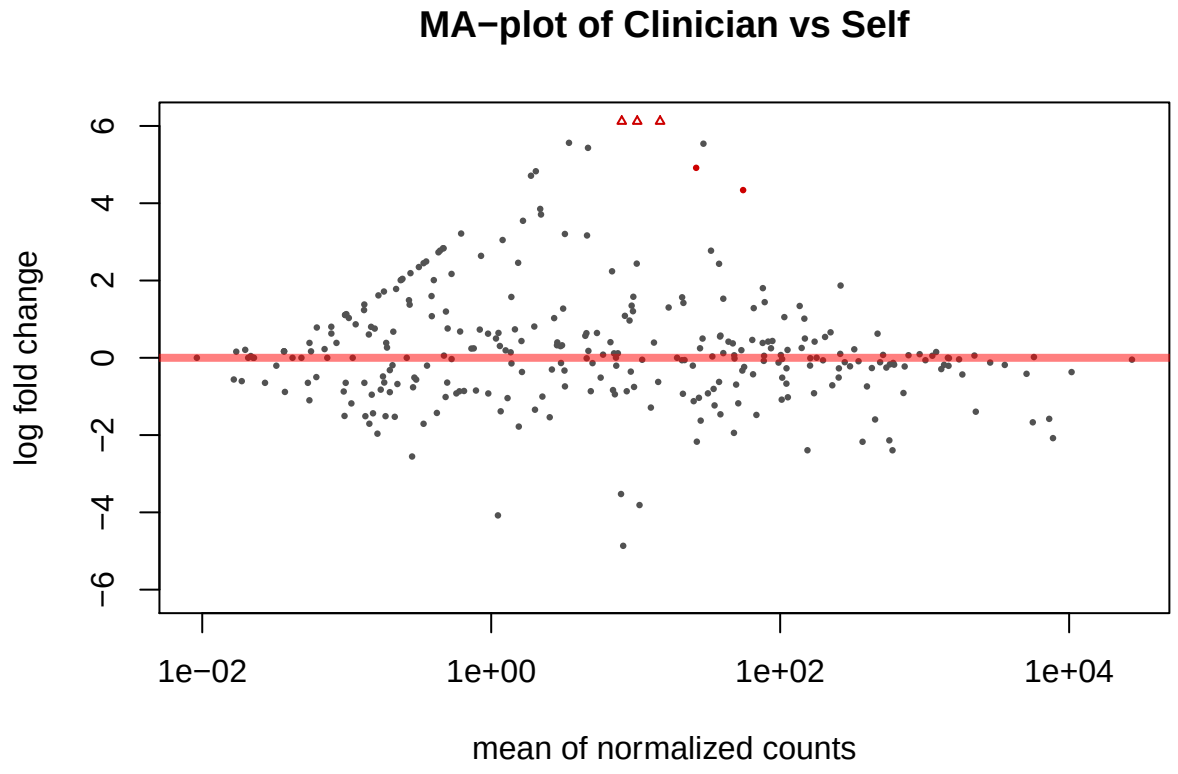
```
## estimating dispersions
```

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

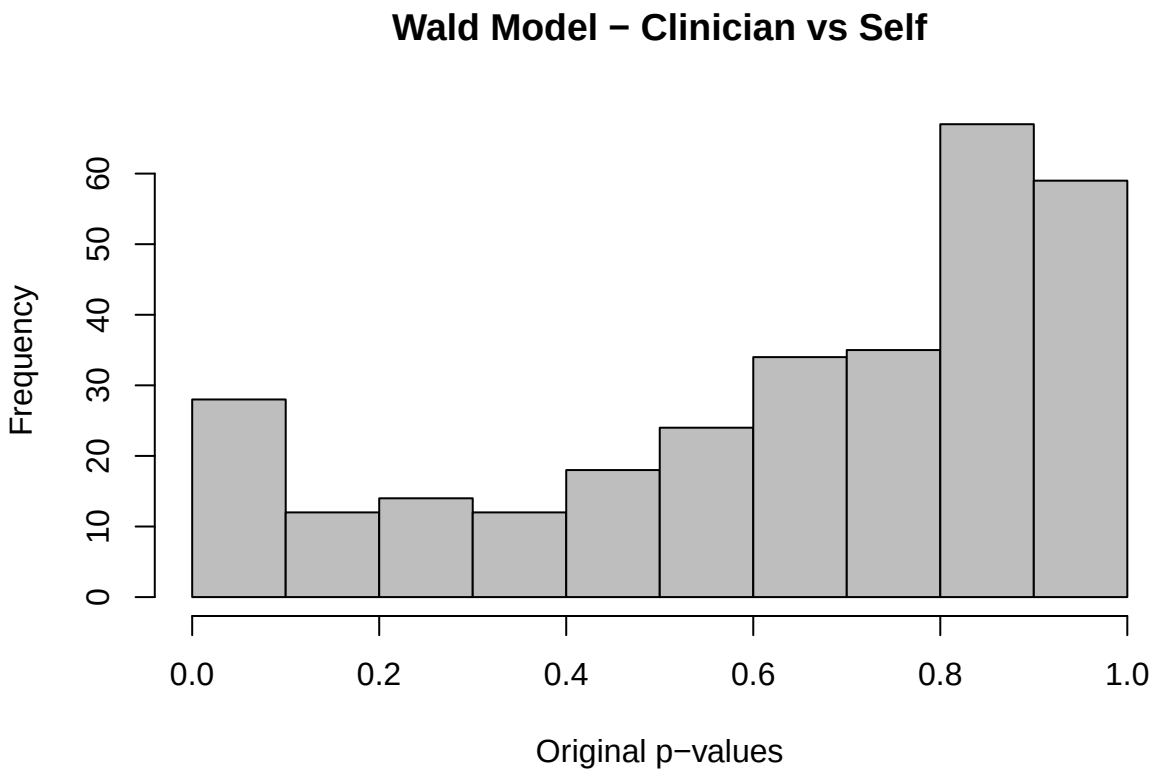
```
# Plot of Dispersion Estimates
plotDispEsts(ds, ylim = c(1e-8, 1e4))
```




```
# 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")
```



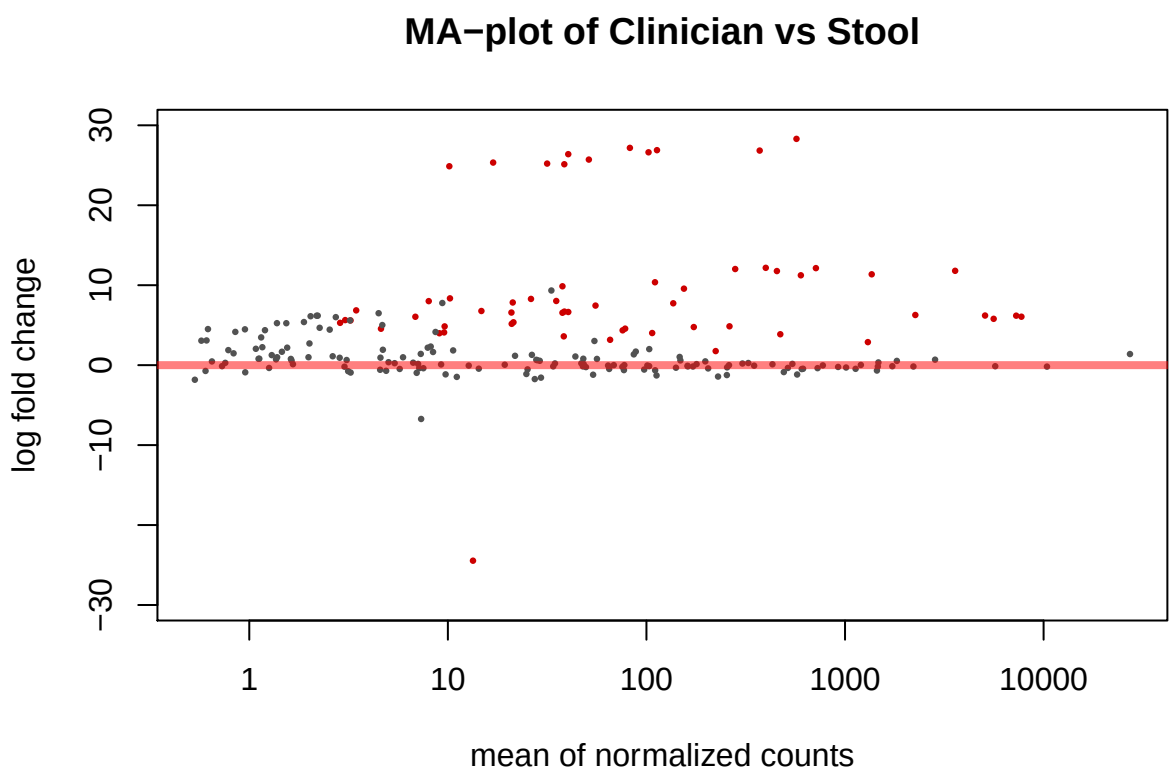

```
hist(resCTST$pvalue, col = "gray", main = "Wald Model - Clinician vs Self", xlab = "Original p-values")
```




```
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
## ASV473  55.398735       4.339790 0.7977869 5.439786 5.334449e-08 8.081691e-06
## ASV930  10.246362       7.072850 1.2770450 5.538450 3.051598e-08 8.081691e-06
## ASV1129  8.010744       6.131688 1.1954151 5.129338 2.907624e-07 2.936700e-05
## ASV658  26.218338       4.915902 1.1727001 4.191952 2.765650e-05 1.764129e-03
## ASV1164 14.753210       6.139523 1.4686762 4.180311 2.911104e-05 1.764129e-03
##          Kingdom          Phylum               Class            Order
## ASV473  Bacteria Proteobacteria Gammaproteobacteria  Pseudomonadales
## ASV930  Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV1129 Bacteria     Firmicutes          Clostridia    Clostridiales
## ASV658  Bacteria Proteobacteria Gammaproteobacteria    Aeromonadales
## ASV1164 Bacteria Proteobacteria Gammaproteobacteria Enterobacterales
##                   Family                      Genus
## ASV473  Pseudomonadaceae                  Pseudomonas
## ASV930    Clostridiaceae  Clostridium_sensu_stricto_5
## 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")
```



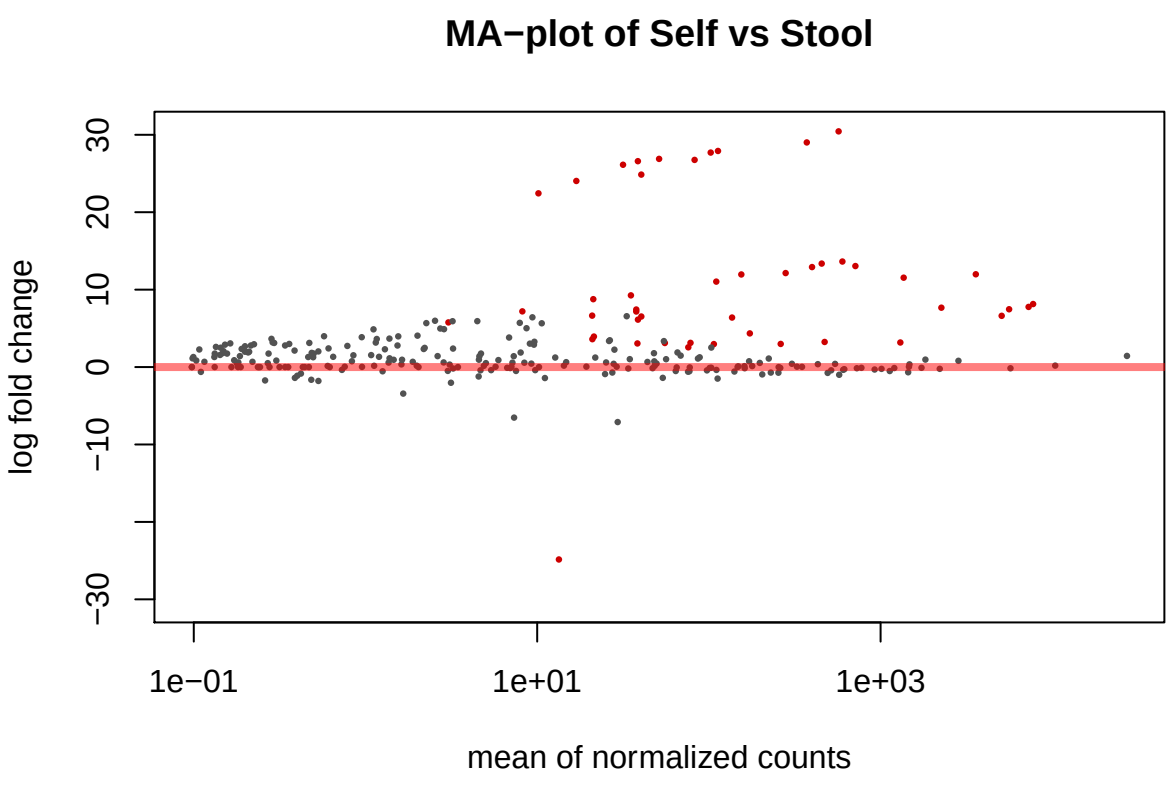

```r
hist(resCTS$pvalue, col = "gray", main = "Wald Model - Clinician vs Stool", xlab = "Original p-values")
```


```
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)
```

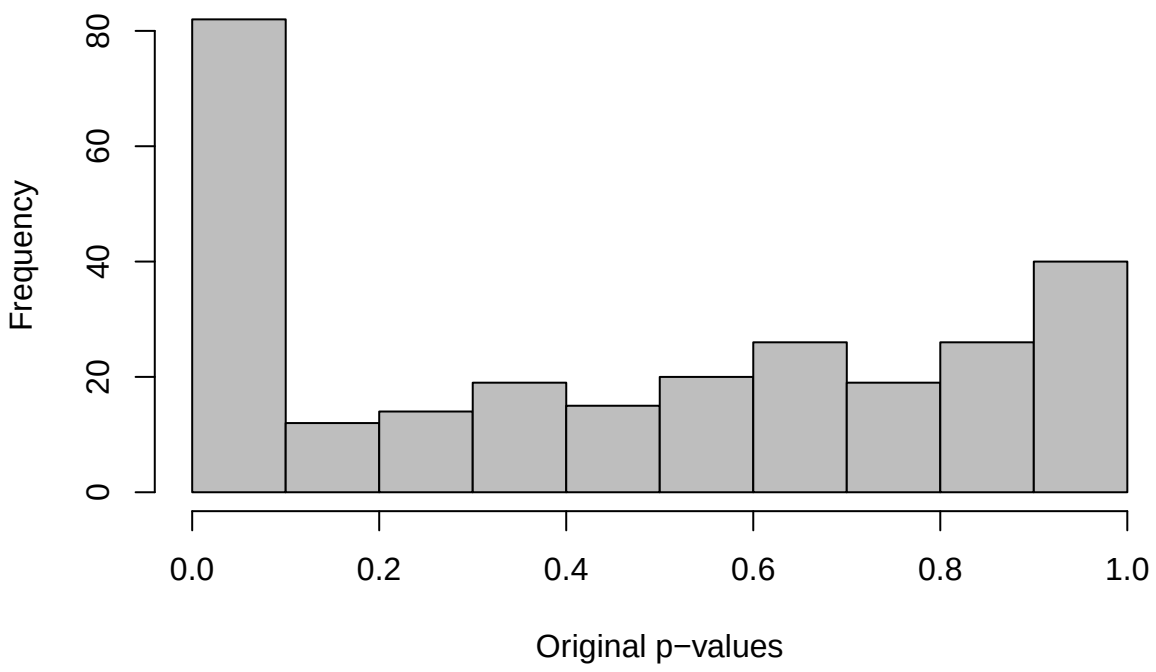
```
##          baseMean log2FoldChange     lfcSE     stat        pvalue          padj
## ASV324  371.68353       26.83942 1.1951058 22.45778 1.074225e-111 2.245130e-109
## ASV262  569.52608       28.30378 1.3237517 21.38149 1.987046e-101  2.076463e-99
## ASV365  102.45995       26.62194 1.5030862 17.71152  3.417104e-70  2.380583e-68
## ASV662  112.98959       26.88915 1.6814757 15.99140  1.467033e-57  7.665245e-56
## ASV283   82.54958       27.17414 1.7632089 15.41175  1.364563e-53  5.703875e-52
## ASV5   3585.31720       11.79910 0.7864744 15.00252  7.068126e-51  2.462064e-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)
```

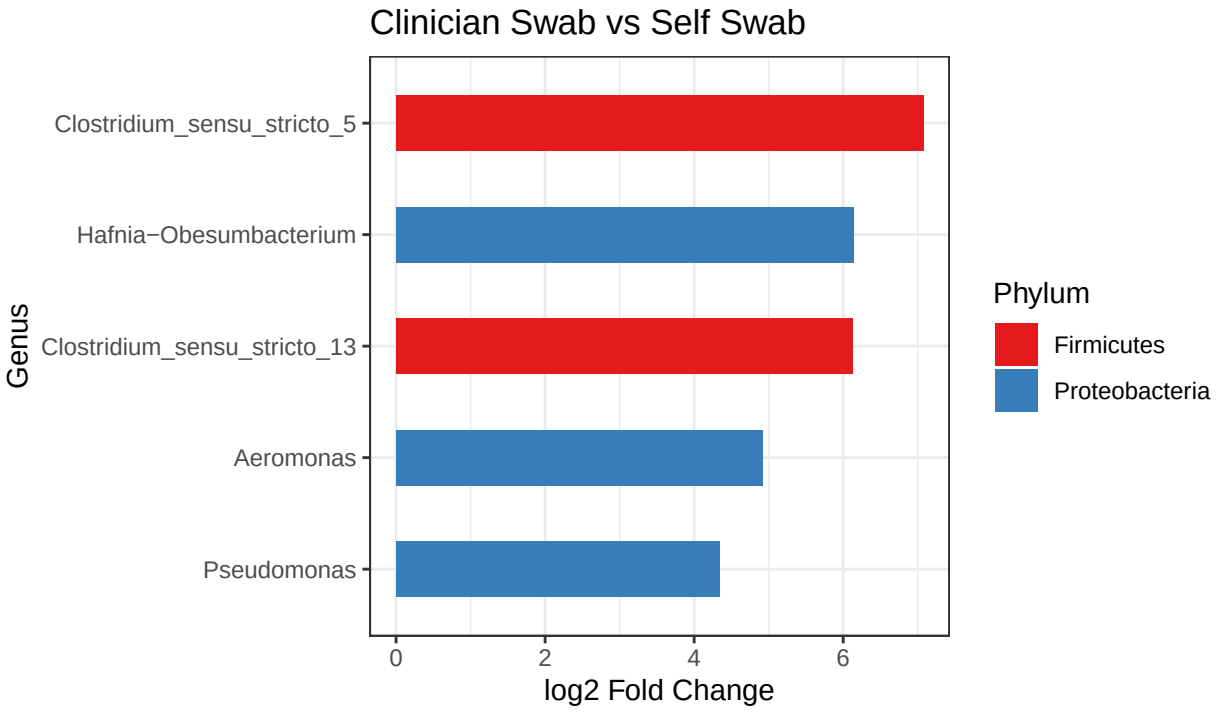
```
##          baseMean log2FoldChange     lfcSE     stat        pvalue          padj
## ASV324  371.68353       29.01285 1.1939499 24.29989 1.965677e-130 5.366298e-128
## ASV262  569.52608       30.44088 1.3230614 23.00791 3.884562e-117 5.302427e-115
## ASV365  102.45995       27.70526 1.5015745 18.45081  5.137379e-76  4.675015e-74
## ASV662  112.98959       27.91069 1.6802657 16.61088  5.813687e-62  3.967841e-60
## ASV5   3585.31720       11.98355 0.7864328 15.23785  1.982790e-52  1.082603e-50
## ASV283   82.54958       26.75502 1.7629576 15.17621  5.082879e-52  2.312710e-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
```

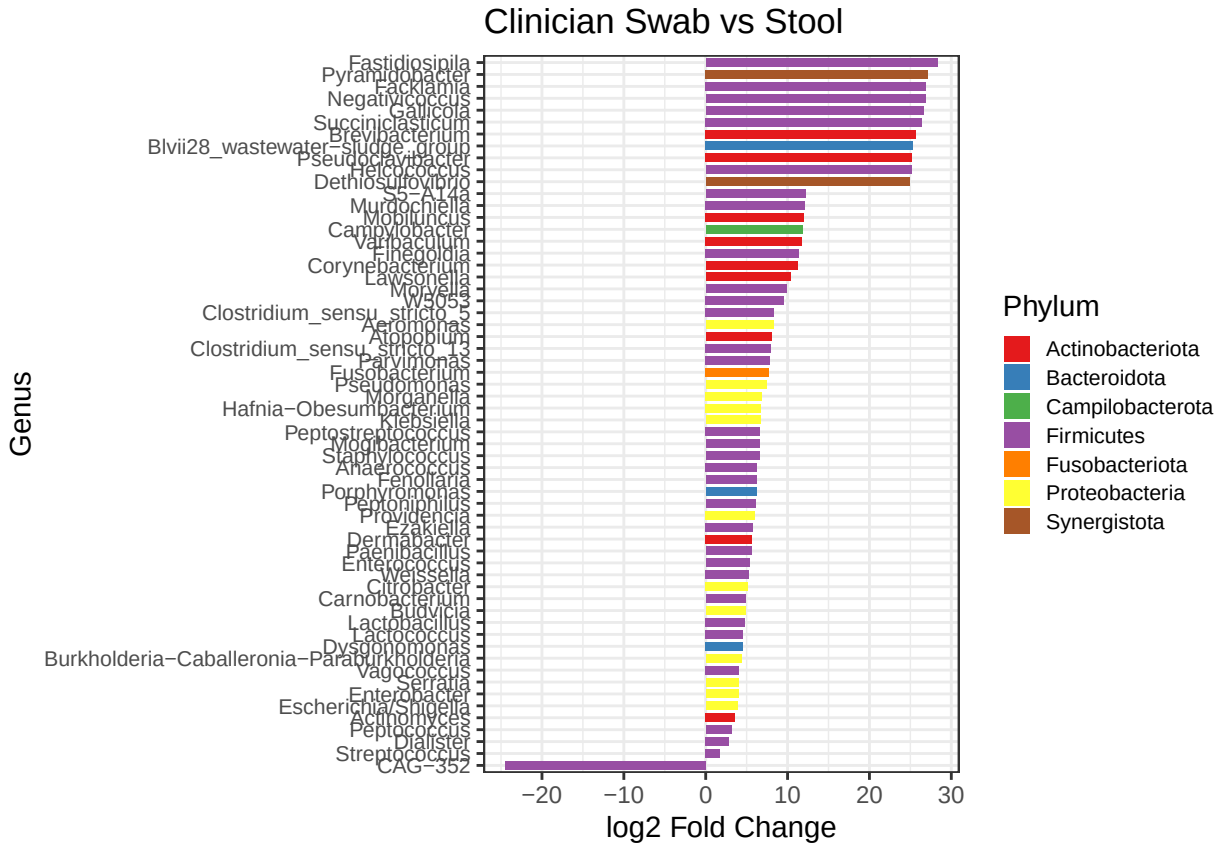
```
# Save .csv of significant fold change results
resCTST_sig$Comparison <- "Clinician Taken Swab vs Self Taken Swab"
resCTS_sig$Comparison <- "Clinician Taken Swab vs Stool"
resSTS_sig$Comparison <- "Self Taken Swab vs Stool"

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

Differential Abundance Figure

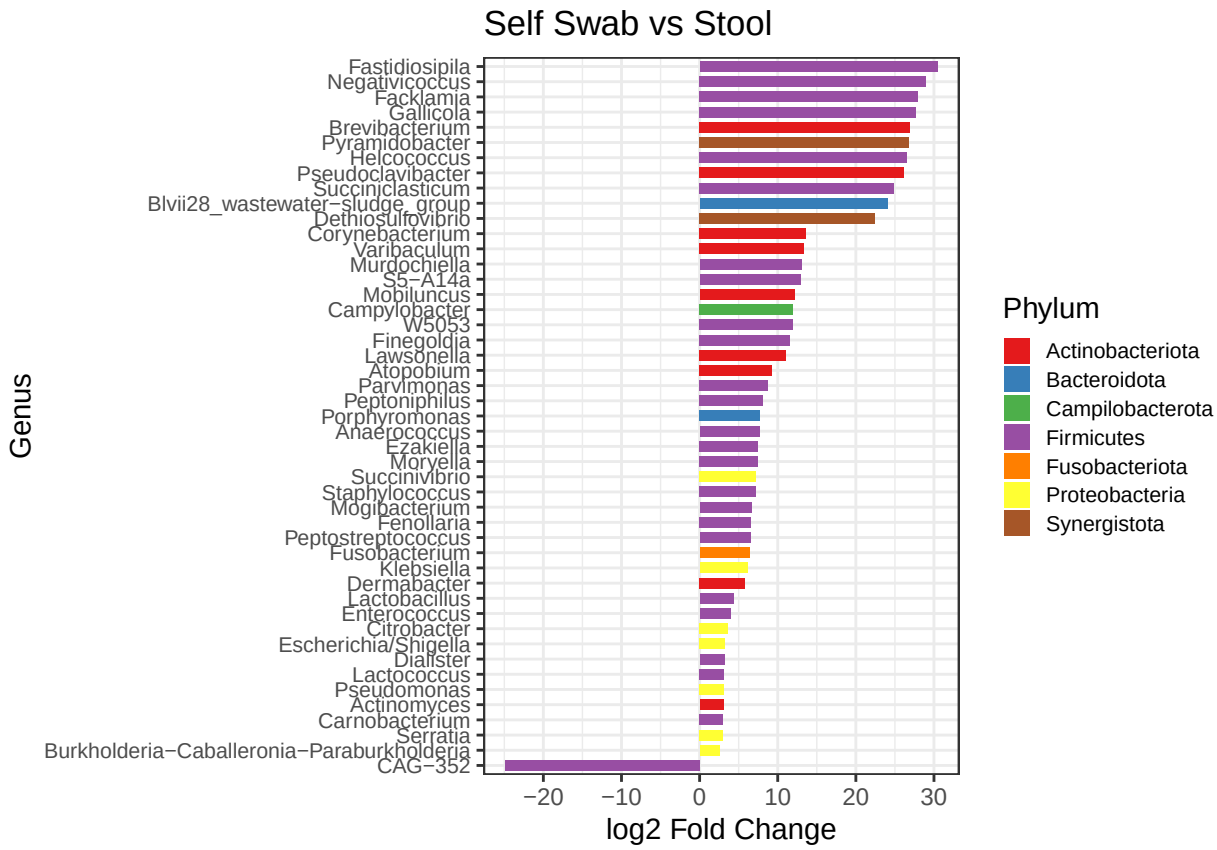
```
ggplot(resCTST_sig, aes(x = log2FoldChange, y = reorder(Genus, log2FoldChange), fill= Phylum)) +
  geom_bar(stat = "identity", position = "identity", width = 0.5) +
  labs(title = "Clinician Swab vs Self Swab", y = "Genus", x = "log2 Fold Change") +
  theme(aspect.ratio = 1) +
  scale_fill_brewer(palette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

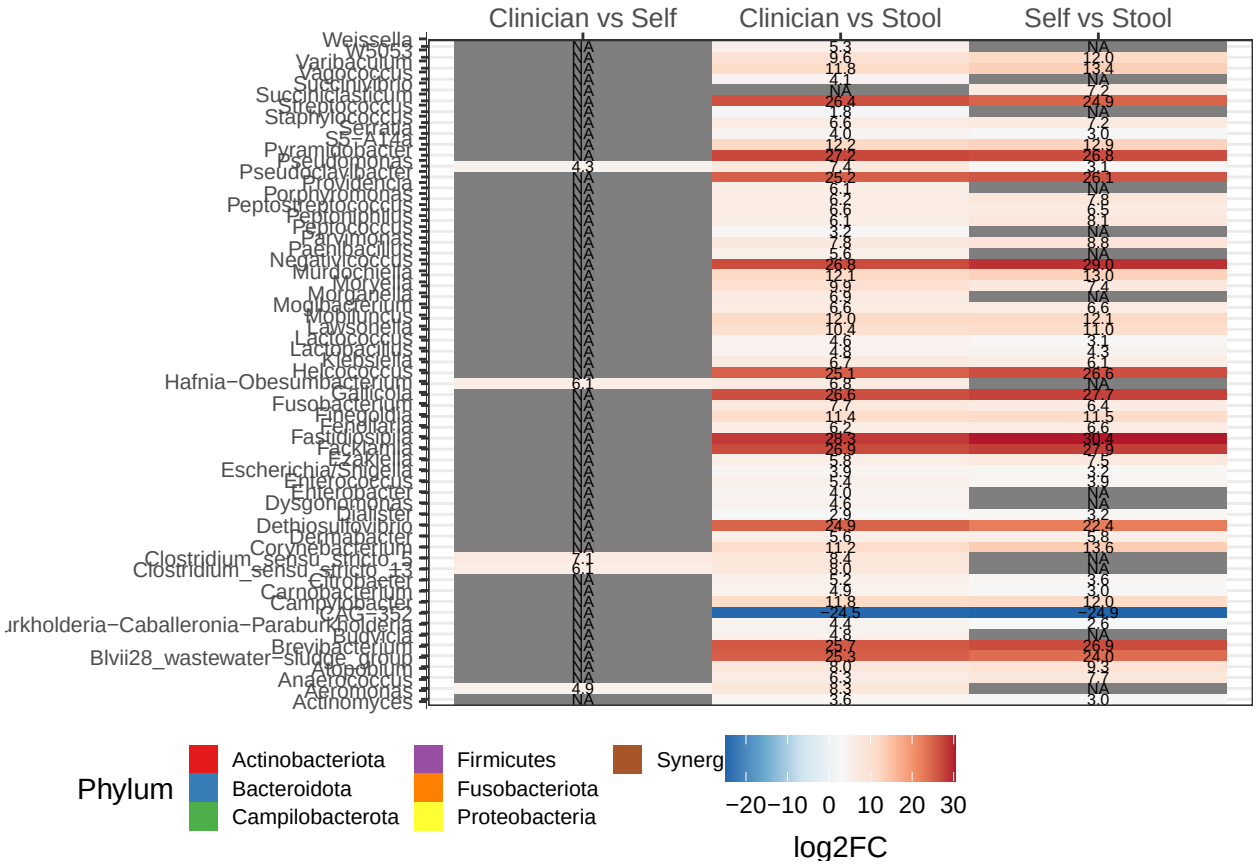

```
ggsave("Purcell Final/Final_results/5)Differential_Abundance_swabsVSstool.pdf", width = 12, height = 8)
```

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 = 13.5, 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,
       device = cairo_pdf)
```

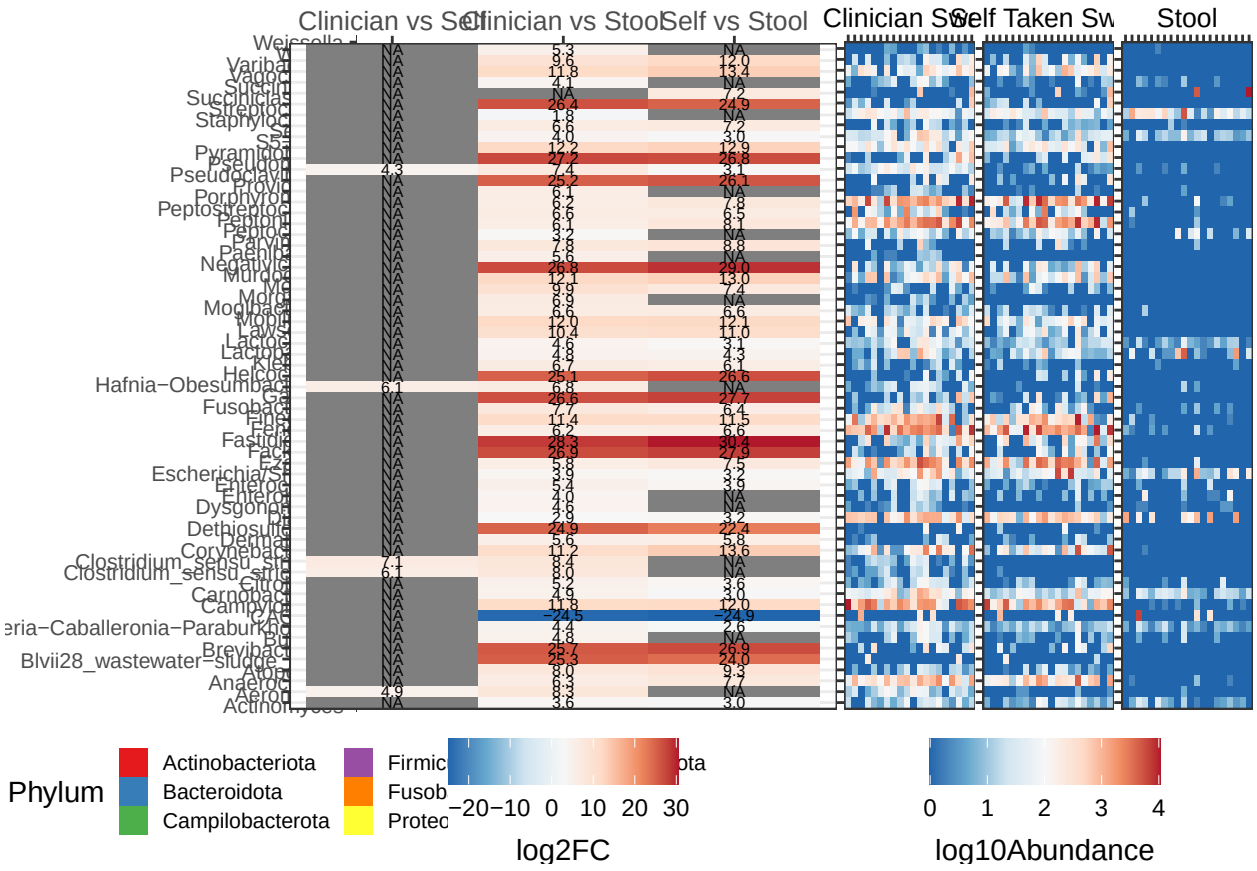
Species Abundance ggplot Heatmap

```
# 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, device = cairo_pdf)
```

Boxplot Sanity Checks

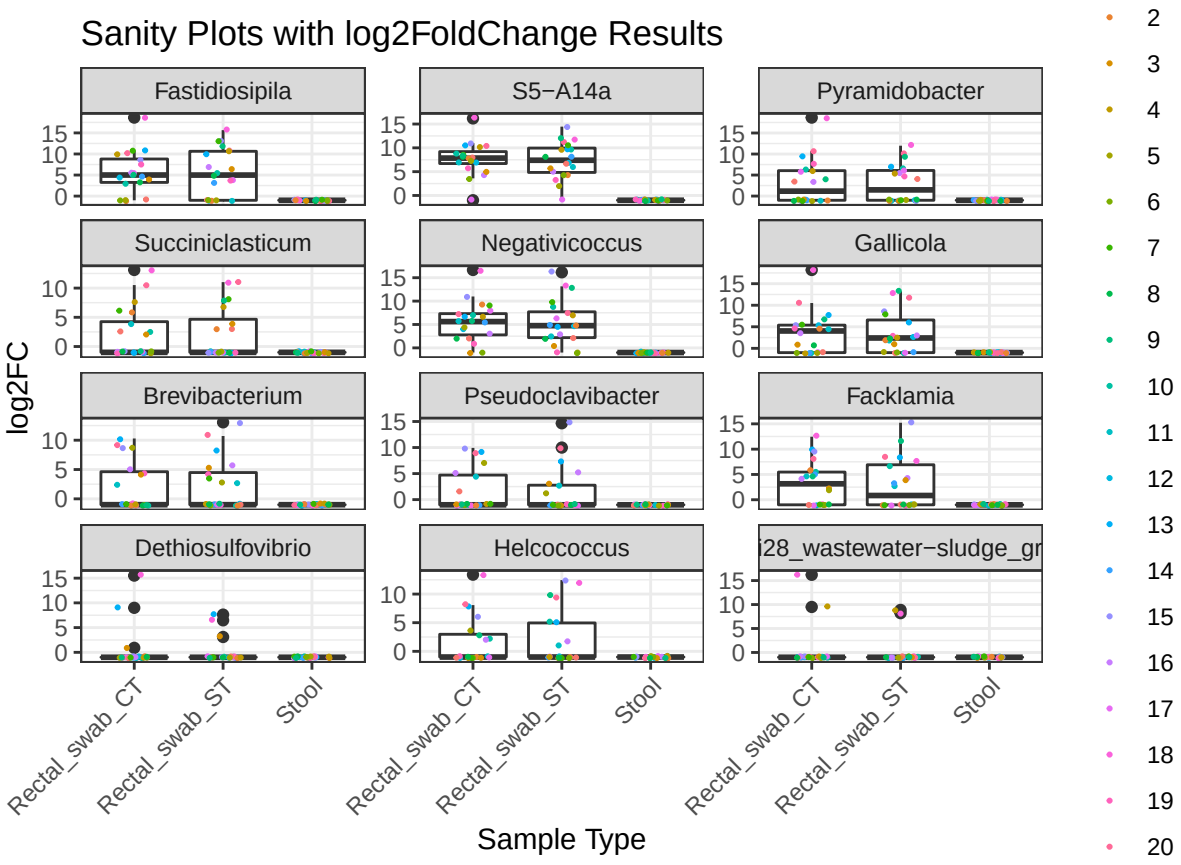
```
resCTS_sig <- resCTS_sig[order(-resCTS_sig$log2FoldChange),]

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))
```

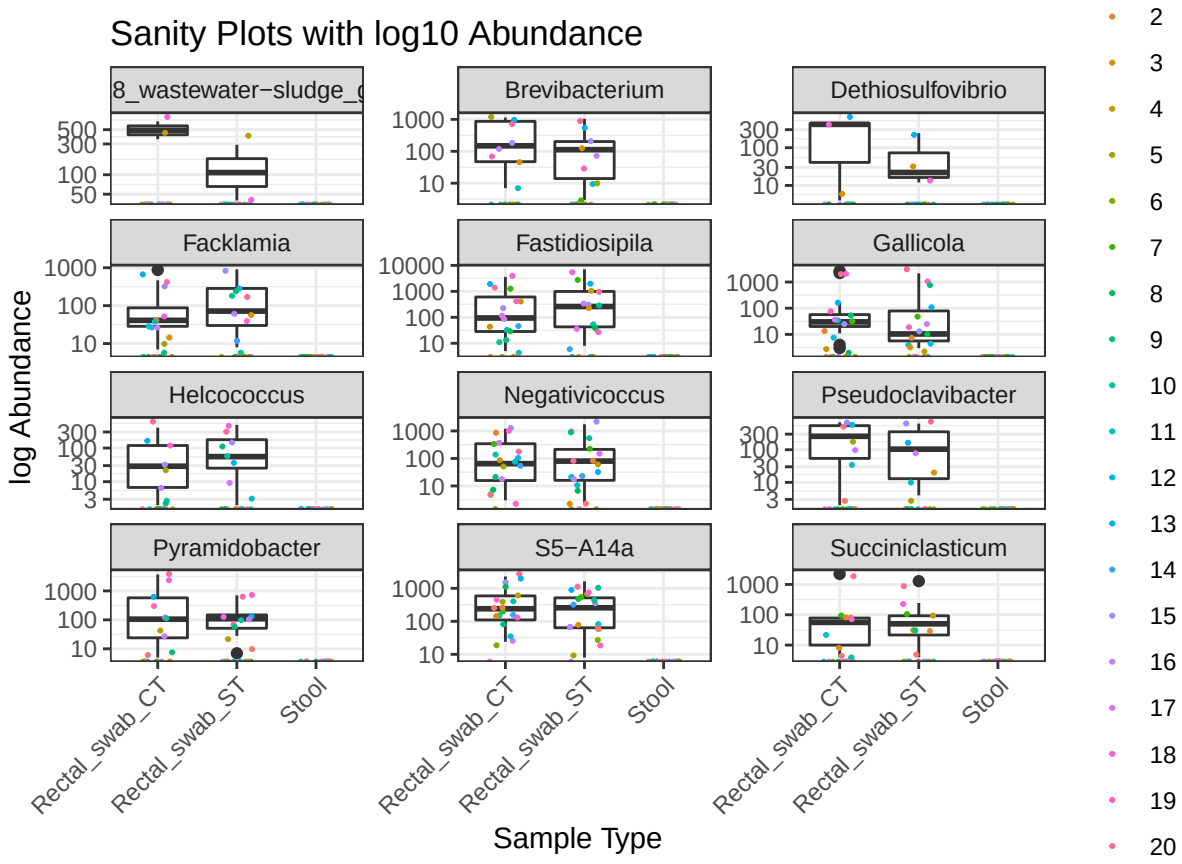
```
## Warning: The labeller API has been updated. Labellers taking `variable` and
## `value` arguments are now deprecated. See labellers documentation.
```




```
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
```