Differential abundance testing with DESeq2

Jacob Westaway

Last updated on 2021-07-31

About

DESeq2 was used to explore potential taxonomic differences between probiotic treatment groups. Continuous predictors were scaled and centered. Multicollinearity was assessed, and collinear variables were not included in the model. Taxa were agglomerated at the genus level, due to the limited taxonomic depth of 16S, and low frequency taxa were removed to only identify clinically relevant differences. A Wald Test with the BH multiple inference correction was performed to obtain taxa that were significantly differentially abundant.

The code to create the initial data objects used in this workflow can be found in the 'Pipeline.Rmd'.

Load packages

```
sapply(c("DESeq2", "phyloseq", "dplyr", "ggplot2", "grid",
         "gridExtra", "ggpubr", "sjPlot", "pheatmap", "tidyverse", "vegan"),
          require, character.only = TRUE)
```

Centre and scale continuous variables.

```
centre_and_scale <- function(data){
# get numeric variables
data2 <- data %>%
  select_if(is.numeric)
# entering and scaling over variables
data3 <- sapply(data2, function(x) scale(x, center=T, scale = 2*sd(x))) %>%
  as.data.frame() %>%
  rownames_to_column("RowID")
# join scaled/centred data to non-numeric data
data %>%
  select_if(negate(is.numeric)) %>%
  rownames_to_column("RowID") %>%
  left_join(data3, by = "RowID") %>%
  select(-RowID)
}

sample_data(ps3) <- sample_data(ps3) %>%
  unclass() %>%
  as.data.frame() %>%
  centre_and_scale() %>%
  mutate("Sample" = ID) %>% # needed to save it back into the original ps object
  column_to_rownames("Sample")
```

Test for multicollinearity

- Define the `corvif()`function that takes metadata and creates a linear model to see if any collinearity exists between variables.
- Then use this function on a defined a vector with all the variables to be included in the model.
- If GVIF $< 3 =$ no collinearity.

```
# define myvif function
myvif <- function(mod) {
  v <- vcov(mod)
  assign <- attributes(model.matrix(mod))$assign
  if (names(coefficients(mod)[1]) == "(Intercept)") {
    v <- v[-1, -1]
    assign <- assign[-1]
  } else warning("No intercept: vifs may not be sensible.")
  terms <- labels(terms(mod))
  n.terms <- length(terms)
  if (n.terms < 2) stop("The model contains fewer than 2 terms")
  if (length(assign) > dim(v)[1] ) {
    diag(tmp_cor)<-0
    if (any(tmp_cor==1.0)){
      return("Sample size is too small, 100% collinearity is present")
    } else {
      return("Sample size is too small")
    }
  }
  R <- cov2cor(v)
  detR <- det(R)
  result <- matrix(0, n.terms, 3)
  rownames(result) <- terms
  colnames(result) <- c("GVIF", "Df", "GVIF^(1/2Df)")
  for (term in 1:n.terms) {
    subs <- which(assign == term)
    result[term, 1] <- det(as.matrix(R[subs, subs])) * det(as.matrix(R[-subs, -subs]))/detR
    result[term, 2] <- length(subs)
  }
  if (all(result[, 2] == 1)) {
    result <- data.frame(GVIF=result[, 1])
  } else {
    result[, 3] <- result[, 1]^(1/(2 * result[, 2]))
  }
  invisible(result)
}

# corvif
corvif <- function(data) {
  data <- as.data.frame(data)
  form    <- formula(paste("fooy ~ ",paste(strsplit(names(data)," "),collapse = " + ")))
  data  <- data.frame(fooy = 1 + rnorm(nrow(data)) ,data)
  lm_mod  <- lm(form,data) # runs linear model with above formula and metadata

  cat("\n\nVariance inflation factors\n\n")
  print(myvif(lm_mod))
}

sample_data(ps3) %>%
  unclass %>%
  as.data.frame() %>%
  select(Feeding_Type, NEC, Sepsis, Mode_of_Delivery, Neonatal_Antibiotics,
         Chorioamnionitis, Preeclampsia, ROP, Primary_Group, Batch) %>%
  centre_and_scale() %>%
  corvif()
```

Perform DESeq2 analysis

- Convert from *phyloseq* to *deseq* object.
- Use prevviously defined functions to calculate geometric means and filter to the most abundant and frequent taxa.
- Use `Deseq()` to perform the normalisation and analysis.
- Extract the results using appropriate previosuly defined function.

```
# define function for Wald test
get_deseq_res_cat <- function(desq_object, contrast_variable, level1, level2){
  res = results(desq_object, contrast = c(contrast_variable, level1, level2))
  res = res[order(res$padj, na.last = NA), ]
  sigtab = res[(res$padj < 0.05), ]
  sigtab = cbind(as(sigtab, "data.frame"),
    as(tax_table(ps3)[rownames(sigtab), ], "matrix"))
  sigtab %>%
  arrange(padj) %>%
  select("log2FoldChange", "lfcSE", "padj", "Genus") %>%
  add_column(Variable = paste0(contrast_variable, level1)) # label the base level
}

phyloseq_to_deseq2(ps3, ~ Primary_Group + Feeding_Type + NEC + Sepsis + Mode_of_Delivery +
                     Neonatal_Antibiotics + Chorioamnionitis + Preeclampsia + ROP +
                     Batch + Diabetes + Antenatal_Antibiotics) %>%
                  calc_geo_means() %>%
                  deseq_filter(10, 10) %>%
                  DESeq(fitType = "local", test = "Wald") %>%
                  get_deseq_res_cat("Primary_Group", "NICU", "SCN") %>%
                  remove_rownames() %>%
                  knitr::kable()
```

log2FoldChange	lfcSE	padj	Genus	Variable
9.448514	2.010470	0.0000339	Enterobacter	Primary_GroupNICU
7.081464	1.866892	0.0006445	Klebsiella	Primary_GroupNICU

log2FoldChange	lfcSE	padj	Genus	Variable
7.769715	2.027889	0.0006445	Veillonella	Primary_GroupNICU
10.050975	2.843655	0.0010621	Escherichia/Shigella	Primary_GroupNICU
-17.574520	4.928732	0.0010621	Rothia	Primary_GroupNICU

Construct histograms to compare pre and post transformation.

- Call `estimateDispersions()` to calculate abundances with `getVarianceStabilizedData()`.
- **NB** piped to `calc_geo_means()` to calculate geometric means and estimate size factors, which is needed for the above.
- **NB.** the samples are in columns in the *deseq* object but in rows for the *phyloseq* object.

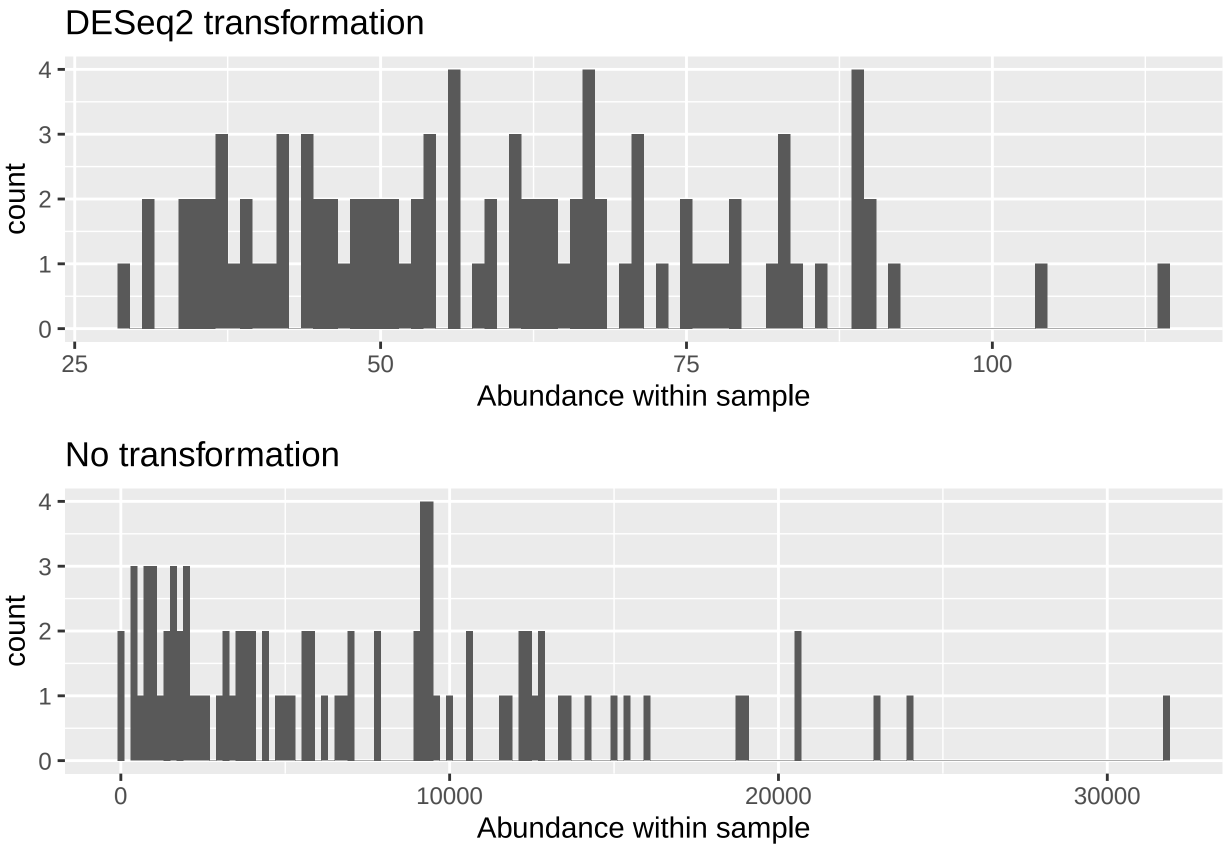
```
# define function to plot transformation
plot_deseq_transformation <- function(deseq_object){
multi.deseq <- estimateDispersions(deseq_object, fitType = "local")

abund_sums_trans <- data.frame(sum = colSums(getVarianceStabilizedData(multi.deseq) ),
                     sample = colnames(getVarianceStabilizedData(multi.deseq) ),
                     type = "DESeq2")

abund_sums_no_trans <- data.frame(sum = rowSums(otu_table(ps3)),
                       sample = rownames(otu_table(ps3)),
                       type = "None")

grid.arrange((ggplot(abund_sums_trans) +
  geom_histogram(aes(x = sum), binwidth = 1) +
  xlab("Abundance within sample") +
  ggtitle("DESeq2 transformation")),
  (ggplot(abund_sums_no_trans) +
  geom_histogram(aes(x = sum), binwidth = 200) +
  xlab("Abundance within sample") +
  ylim(0,4) +
  ggtitle("No transformation")),
  nrow = 2)
}

phyloseq_to_deseq2(ps3, ~ Primary_Group + Feeding_Type) %>%
  calc_geo_means() %>% plot_deseq_transformation()
```



DESeq2 transformation

No transformation

Figure 1: Pre and post transformation of taxonomic counts with DESeq2