Differential abundance testing with DESeq2

Jacob Westaway

Last updated on 2021-08-15

About

DESeq2 was used to explore potential taxonomic differences between covariates. 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'.

Packages

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

Subset data to Admission samples only

- and scale continuous variables.

```
ps.NICU <- subset_samples(ps3, Primary_Group == "NICU" &
        Type == "Admission")

sample_data(ps.NICU) <- sample_data(ps.NICU) %>%
  unclass() %>%
  as.data.frame() %>%
  filter(!is.na(Days_on_antibiotics)) %>% # omit NA samples for modelling
  filter(!is.na(Days_since_birth)) %>%
  centre_and_scale() %>%
  mutate("Sample" = Label) %>% # redo rownames to save it back into the original ps object
  column_to_rownames("Sample")
```

Testing 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.

```
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 <- 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))
}

model_data <- sample_data(ps.NICU) %>%
  unclass() %>%
  as.data.frame()
```

```
model_data <- cbind(model_data$Mode.of.Delivery, model_data$Feeding.Type,
                    model_data$Gestation_Days_scaled, model_data$NEC,
                    model_data$Sepsis, model_data$Chorioamnionitis,
                    model_data$Preeclampsia, model_data$ROP,
                    model_data$Gest_at_collection, model_data$Days_on_antibiotics)

corvif(model_data)
```

Convert from *phyloseq* to *deseq* object

```
multi.deseq = phyloseq_to_deseq2(ps.NICU, ~Sepsis + Feeding.Type +
                                   Chorioamnionitis + Mode.of.Delivery +
                                   Gestational.Age.at.Birth  + NEC +
                                   Preeclampsia + ROP + Gest_at_collection +
                                   Days_on_antibiotics)
```

Calculate geometric means, estimate size factors and filter on frequencu and abundance

- Define function for calculating geometric means.
- Subset out taxa with small counts and low occurance (at least 10 in 20 or more samples).

```
gm_mean = function(x, na.rm = TRUE){
  exp(sum(log(x[x > 0]), na.rm = na.rm) / length(x))
}

geoMeans <- apply(counts(multi.deseq), 1, gm_mean)
multi.deseq <- estimateSizeFactors(multi.deseq, geoMeans = geoMeans)

nc <- counts(multi.deseq, normalized = TRUE)
filtered <- rowSums(nc >= 10) >= 20
multi.deseq <- multi.deseq[filtered,]
```

- Construct histograms to compare pre and post transformation.
- Call `estimateDispersions()` to calculate abundances with `getVarianceStabilizedData()`.
- **NB.** the samples are in columns in the *deseq* object but in rows for the *phyloseq* object.
- Axis adujsted for what best represents the distribution.

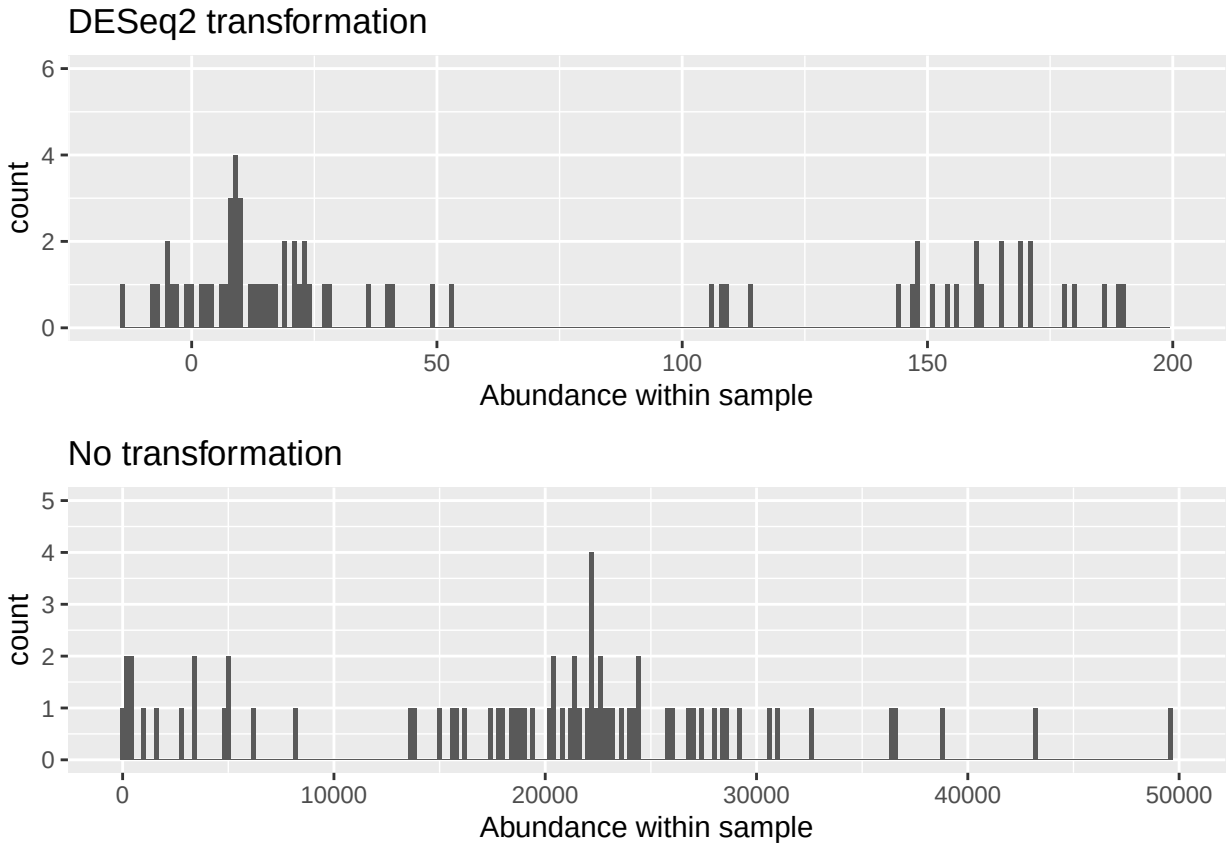
```
multi.deseq <- estimateDispersions(multi.deseq, fitType = "local")

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(ps.NICU)),
                       sample = rownames(otu_table(ps.NICU)),
                       type = "None")

grid.arrange((ggplot(abund_sums_trans) +
```

```
  geom_histogram(aes(x = sum), binwidth = 1) +
  xlab("Abundance within sample") +
  xlim(NA, 200) +
  ylim(0,6) +
  ggtitle("DESeq2 transformation")),
  (ggplot(abund_sums_no_trans) +
  geom_histogram(aes(x = sum), binwidth = 200) +
  xlab("Abundance within sample") +
  ylim(0,5) +
  ggtitle("No transformation")),
  nrow = 2)
```



Statisical test: calculate differential abundances with *DESeq2*.

- Use `Deseq()` to perform the normalisation and analysis.

```
multi.deseq = DESeq(multi.deseq, fitType = "local", test = "Wald")
```

- Define functions to extract the results.
- Extract the results, order by p value, selects significant (<0.05) results, binds this data to the *tax_table* from the *phyloseq* object to get the taxonomic information, and then select and order the desired columns.

```
get_deseq_res <- function(desq_object, contrast_variable, level1, level2){
  res = results(desq_object, contrast = c(contrast_variable, level1, level2))
  res = res[order(res$padj, na.last = NA), ]
  alpha = 0.05
  sigtab = res[(res$padj < alpha), ]
  sigtab = cbind(as(sigtab, "data.frame"),
          as(tax_table(ps.NICU)[rownames(sigtab), ], "matrix"))
  sigtab %>%
  arrange(log2FoldChange) %>%
  select("log2FoldChange", "lfcSE", "padj", "Genus") %>%
  add_column(Variable = paste0(contrast_variable, ":Yes"))
}

get_deseq_res_cont <- function(deseq_object, contrast_variable){
  res = results(deseq_object, name = contrast_variable)
  res = res[order(res$padj, na.last = NA), ]
  sigtab = res[(res$padj < 0.05), ]
  sigtab = cbind(as(sigtab, "data.frame"),
          as(tax_table(ps.NICU)[rownames(sigtab), ], "matrix"))
  sigtab %>%
  arrange(padj) %>%
  select("log2FoldChange", "lfcSE", "padj", "Genus")
}
```

- Use the `get_deseq_res()` to create a table of each of the significant variables.

```
sigtab_admission <- bind_rows(get_deseq_res(multi.deseq,
                              "Chorioamnionitis", "Yes", "No"),
                              get_deseq_res(multi.deseq,
                              "Sepsis", "Yes", "No"),
                              get_deseq_res(multi.deseq,
                              "NEC", "Yes", "No"),
                              get_deseq_res(multi.deseq,
                              "ROP", "Yes", "No")) %>%
                    add_column(Sample = "Admission")
```

Subset data to Discharge samples only

- repeat steps above.

```
ps.NICU <- subset_samples(ps3, Primary_Group == "NICU" &
        Type == "Discharge")

sample_data(ps.NICU) <- sample_data(ps.NICU) %>%
  unclass() %>%
  as.data.frame() %>%
  filter(!is.na(Days_on_antibiotics)) %>% # omit NA samples for modelling
  filter(!is.na(Days_since_birth)) %>%
  centre_and_scale() %>%
  mutate("Sample" = Label) %>% # redo rownames to save it back into the original ps object
  column_to_rownames("Sample")
```

```
multi.deseq = phyloseq_to_deseq2(ps.NICU, ~Sepsis + Feeding.Type +
                                   Chorioamnionitis + Mode.of.Delivery +
                                   Gestational.Age.at.Birth  + NEC +
                                   Preeclampsia + ROP + Gest_at_collection +
                                   Days_on_antibiotics)

geoMeans <- apply(counts(multi.deseq), 1, gm_mean)
multi.deseq <- estimateSizeFactors(multi.deseq, geoMeans = geoMeans)

nc <- counts(multi.deseq, normalized = TRUE)
filtered <- rowSums(nc >= 10) >= 20
multi.deseq <- multi.deseq[filtered,]

multi.deseq = DESeq(multi.deseq, fitType = "local", test = "Wald")
multi.deseq.clean <- multi.deseq[which(mcols(multi.deseq)$betaConv),]

sigtab_discharge <- bind_rows(get_deseq_res(multi.deseq,
                              "Preeclampsia", "Yes", "No"),
                              get_deseq_res(multi.deseq,
                              "Feeding.Type", "Breastmilk", "Formula"),
                              get_deseq_res(multi.deseq,
                              "Feeding.Type", "Formula", "Breastmilk and Formula")) %>%
                    add_column(Sample = "Discharge") %>%
                    select(-Variable) %>%
                    add_column(Variable = c("Preeclampsia:Yes", "Diet:Breastmilk",
                                            "Diet:Breastmilk", "Diet:Breastmilk",
                                            "Diet:Formula"))
```

Merge *Admission* and *Discharge* outputs

```
bind_rows(sigtab_admission, sigtab_discharge) %>%
  remove_rownames() %>%
  kable()
```

log2FoldChange	lfcSE	padj	Genus	Variable	Sample
3.090225	0.9860045	0.0362017	Staphylococcus	Chorioamnionitis:Yes	Admission
-17.584377	3.2160963	0.0000010	Enhydrobacter	Sepsis:Yes	Admission
-15.384213	3.9544329	0.0010510	Pseudomonas	Sepsis:Yes	Admission
10.326119	2.8388170	0.0019273	Bifidobacterium	Sepsis:Yes	Admission
-11.623161	2.3100410	0.0000102	Bifidobacterium	NEC:Yes	Admission
4.843475	0.9987588	0.0000260	Staphylococcus	ROP:Yes	Admission
-27.647935	2.6395415	0.0000000	Escherichia/Shigella	Preeclampsia:Yes	Discharge
-4.252449	1.6658178	0.0330475	Veillonella	Diet:Breastmilk	Discharge
2.536712	0.8716862	0.0289032	Bifidobacterium	Diet:Breastmilk	Discharge
3.643593	1.4569958	0.0330475	Klebsiella	Diet:Breastmilk	Discharge
-5.519725	1.9089134	0.0306674	Lactobacillus	Diet:Formula	Discharge

Finished