Generalised linear mixed effects modelling

Jacob Westaway

Last updated on 2021-07-31

About

To examine the effect of probiotic-treatment on Shannon diversity, a generalised linear mixed effects regression model was created using lme4. Shannon diversity was calculated at the ASV level (normalised through TSS), and continuous predictors were scaled and centered. Multicollinearity was assessed with the AED package, and collinear variables were removed. To control for high inter-individual variation in the preterm infant microbiome, the infants identification was included as a random factor.

After creation of the initial model with lme4, backwards selection was used to find the least complex, yet adequate, model by comparing Akaike's Information Criterion (AIC) scores and removing predictors that did not contribute to variation in the model. A post-hoc pairwise Tukey comparison (correcting for multiple comparisons) was used to assess the effect of probiotic-treatment on alpha diveristy using the emmeans package.

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

Packages

```
sapply(c("phyloseq", "tidyverse", "knitr", "lme4", "emmeans", "MuMIn", "aods3", "sjPlot"),
        require, character.only = TRUE)
```

Calculate alpha diversity

```
# define calc_alpha_diversity function
calc_alpha_diversity <- function(ps2){
# calculate metrics
ps_alpha_div <- ps2 %>%
                estimate_richness(measures = c("Shannon", "Observed", "Chao1")) %>%
                select(-se.chao1)

# creat ID column based on rownames
ps_alpha_div <- rownames_to_column(ps_alpha_div, var = "ID") %>%
                mutate(ID = as.factor(gsub("X", "", ID)))

# join alpha metrics with metadata by the ID column
Metadata %>%
  filter(Type == "Discharge") %>%
```

```r
  right_join(ps_alpha_div, by = "ID") %>%
  as.data.frame()
}

ps_metadata <- calc_alpha_diversity(ps2)
```

Centre and scale data

```r
# define centre and scale function
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)
}

glm_data <- ps_metadata %>%
  mutate(Shannon = as.factor(Shannon)) %>%
  centre_and_scale() %>%
  mutate(Shannon = as.character(Shannon)) %>%
  mutate(Shannon = as.numeric(Shannon))
```

Test for collinearity using known microbiome-covariates

```r
# defin 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))
}

glm_data %>%
  select(Primary_Group, Feeding_Type, NEC, Sepsis, Mode_of_Delivery,
         Neonatal_Antibiotics, Chorioamnionitis, Preeclampsia, ROP,
         Batch, Diabetes ,Antenatal_Antibiotics) %>%
  corvif()
```

Fit Model

```
global <- lme4::glmer(Shannon ~ Primary_Group + Feeding_Type + NEC + Sepsis +
                        Mode_of_Delivery + Neonatal_Antibiotics + Chorioamnionitis +
                        Preeclampsia + ROP + Batch + Diabetes + Antenatal_Antibiotics +
                        (1|URN), data = (glm_data %>% filter(Shannon > 0)),
                        family = Gamma(link = "log"))
global
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula:
## Shannon ~ Primary_Group + Feeding_Type + NEC + Sepsis + Mode_of_Delivery +
##     Neonatal_Antibiotics + Chorioamnionitis + Preeclampsia +
##     ROP + Batch + Diabetes + Antenatal_Antibiotics + (1 | URN)
##    Data: (glm_data %>% filter(Shannon > 0))
##      AIC      BIC   logLik deviance df.resid
## 101.0538 141.2276 -34.5269  69.0538       75
## Random effects:
##  Groups   Name        Std.Dev.
##  URN      (Intercept) 0.7842
##  Residual             0.2959
## Number of obs: 91, groups:  URN, 84
## Fixed Effects:
##                        (Intercept)                    Primary_GroupSCN
##                            0.33988                            -1.38605
## Feeding_TypeBreastmilk and Formula                 Feeding_TypeFormula
##                            0.14679                             0.63682
##                             NECYes                           SepsisYes
##                           -0.86733                            -0.67611
##            Mode_of_DeliveryVaginal             Neonatal_AntibioticsYes
##                            0.20867                            -0.67071
##                ChorioamnionitisYes                     PreeclampsiaYes
##                           -0.51670                            -0.69769
##                             ROPYes                           BatchRun2
##                            0.42805                             1.94270
##                        DiabetesYes            Antenatal_AntibioticsYes
##                           -0.06906                            -0.02022
## convergence code 0; 0 optimizer warnings; 1 lme4 warnings
```

```
gof(global)
r.squaredGLMM(global)
```

Backwards Selection

```
dfun <- function(x) {
  x$AIC <- x$AIC-min(x$AIC)
  names(x)[2] <- "dAIC"
  x
}

dfun(drop1(global))
```

```
## Single term deletions
##
## Model:
## Shannon ~ Primary_Group + Feeding_Type + NEC + Sepsis + Mode_of_Delivery +
##     Neonatal_Antibiotics + Chorioamnionitis + Preeclampsia +
##     ROP + Batch + Diabetes + Antenatal_Antibiotics + (1 | URN)
##                        npar    dAIC
## <none>                      1.9879
## Primary_Group            1  9.5572
## Feeding_Type             2  0.6946
## NEC                      1  1.1860
## Sepsis                   1  0.2102
## Mode_of_Delivery         1  0.4770
## Neonatal_Antibiotics     1  2.6833
## Chorioamnionitis         1  1.8770
## Preeclampsia             1  3.2534
## ROP                      1  1.3745
## Batch                    1 17.5078
## Diabetes                 1  0.0189
## Antenatal_Antibiotics    1  0.0000
```

```
global2 <- lme4::glmer(Shannon ~ Primary_Group +  Chorioamnionitis +
          Preeclampsia + Batch + (1|URN),
          data = (glm_data %>% filter(Shannon > 0)), family = Gamma(link = "log"))

dfun(drop1(global2))
```

```
## Single term deletions
##
## Model:
## Shannon ~ Primary_Group + Chorioamnionitis + Preeclampsia + Batch +
##     (1 | URN)
##                   npar    dAIC
## <none>                 0.0000
## Primary_Group       1  8.8658
## Chorioamnionitis    1  0.8190
## Preeclampsia        1  0.4345
## Batch               1 12.9343
```

Calculate the statistical pairwise/adjusted significance

```
emmeans(global2, list(pairwise ~ Primary_Group), adjust = "tukey") %>%
  pairs() %>%
  as.data.frame()
```

```
##   emmeans.of.Primary_Group.contrast emmeans.of.Primary_Group.estimate
## 1                        NICU - SCN                          1.406928
##   emmeans.of.Primary_Group.SE emmeans.of.Primary_Group.df
## 1                   0.4058227                         Inf
##   emmeans.of.Primary_Group.z.ratio emmeans.of.Primary_Group.p.value
## 1                         3.466855                     0.0005265859
##   pairwise.differences.of.Primary_Group.contrast
## 1                                      (nothing)
##   pairwise.differences.of.Primary_Group.estimate
## 1                                             NA
##   pairwise.differences.of.Primary_Group.SE
## 1                                       NA
##   pairwise.differences.of.Primary_Group.df
## 1                                       NA
##   pairwise.differences.of.Primary_Group.z.ratio
## 1                                            NA
##   pairwise.differences.of.Primary_Group.p.value
## 1                                            NA
```

Calculate the goodness of fit and R2.

```
gof(global2)
```

```
## D  = 2.0534, df = 84, P(>D) = 1
## X2 = 1.9356, df = 84, P(>X2) = 1
```

```
r.squaredGLMM(global2)
```

```
##                 R2m       R2c
## delta     0.3073220 0.9200794
## lognormal 0.3083756 0.9232338
## trigamma  0.3061796 0.9166595
```

Plot

```
plot_model(global2, vline.color = "red",
                   sort.est = TRUE, type="pred",
                   terms = "Primary_Group", title = "", colors = "v",
                   axis.title = c("Probiotic-treatment", "Alpha Diversity")) +
  font_size(axis_title.x = 30, axis_title.y = 30, labels.x = 30, labels.y = 30)
```

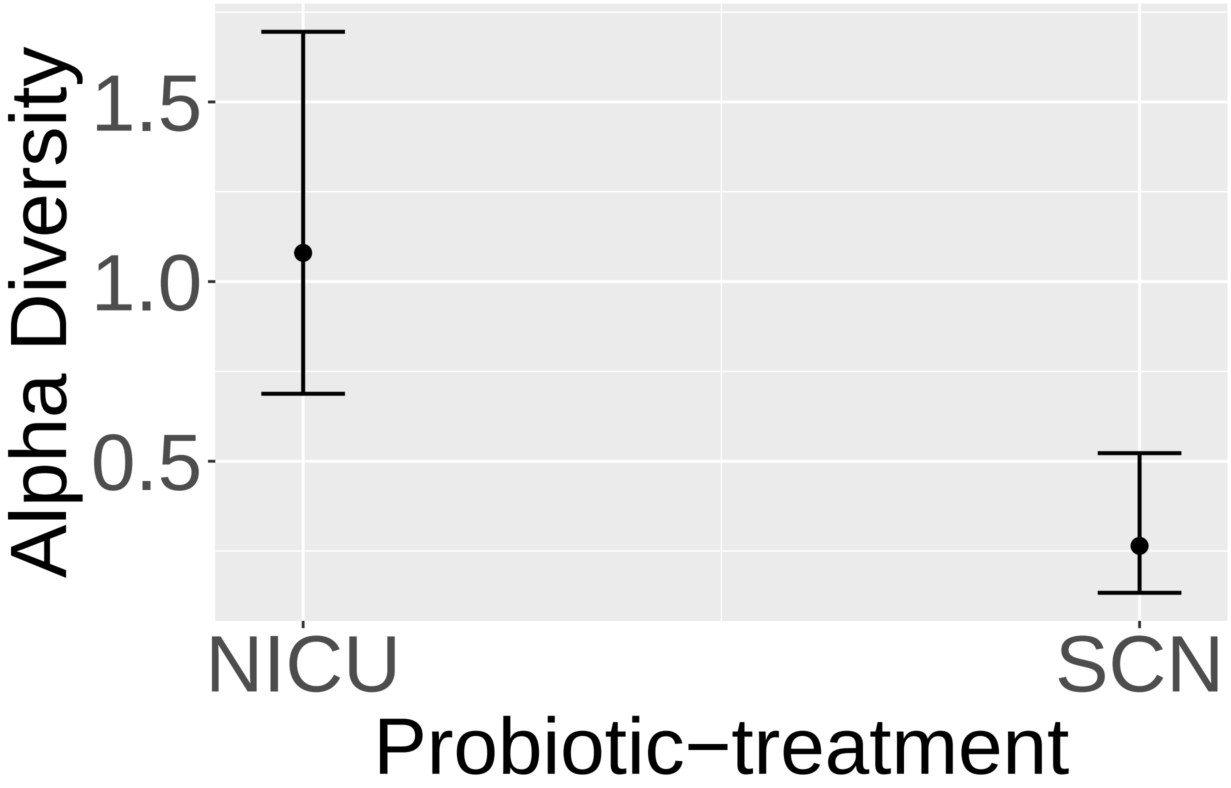

Figure 1: Dot and whisker plot of the estimates for probiotic-treatment generlaised linear mixed effects model results (NICU = probiotic-treated, SCN = Non-treated)