Supplementary SD3: Human-provisioned foods reduce gut microbiome diversity in American black bears

Sierra J. Gillman, Erin A. McKenney, Diana J.R. Lafferty

1/23/2021

All Data Files necessary for replication of the analysis along with Demultiplexed Sequences can be found in a Zenodo.org ***repository: 10.5281/zenodo.5133808***

Supplemental Code

Data Cleaning and Normalization

Import and clean data

Package qiime2R

```r
library(ggplot2)
library(microbiome)
library(phyloseq)
library(qiime2R)
library(picante)
library(brms)
library(dplyr)
library(coda)
library(Rmisc)
library(tidyverse)
library(cowplot)
library(knitr)
library(bookdown)
library(car)
library(Rmisc)
library(RVAideMemoire)
library(kableExtra)
library(SRS)

setwd("~/Desktop/Projects/Bear/Bear-R/CLEAN/FINAL")
```

```r
## import artifacts & metadata file
metadata<-read_tsv("Metafile.tsv")
SVs<-read_qza("table.qza")
taxonomy<-read_qza("taxonomy.qza")
taxtable<-taxonomy$data %>%
  as_tibble() %>%
  separate(Taxon, sep=";", c("Domain",
                             "Phylum", "Class",
                             "Order", "Family",
                             "Genus", "Species"))
tree<-read_qza("rooted-tree.qza")


## Create the phyloseq object
phy_obj<-phyloseq(
  otu_table(SVs$data, taxa_are_rows = T),
  phy_tree(tree$data),
  tax_table(as.data.frame(taxtable) %>%
              select(-Confidence) %>%
              column_to_rownames("Feature.ID") %>%
              as.matrix()), #moving the taxonomy to the way phyloseq wants it
  sample_data(metadata %>%
                as.data.frame() %>%
                column_to_rownames("SampleID")))
```

Clean up taxonomy names

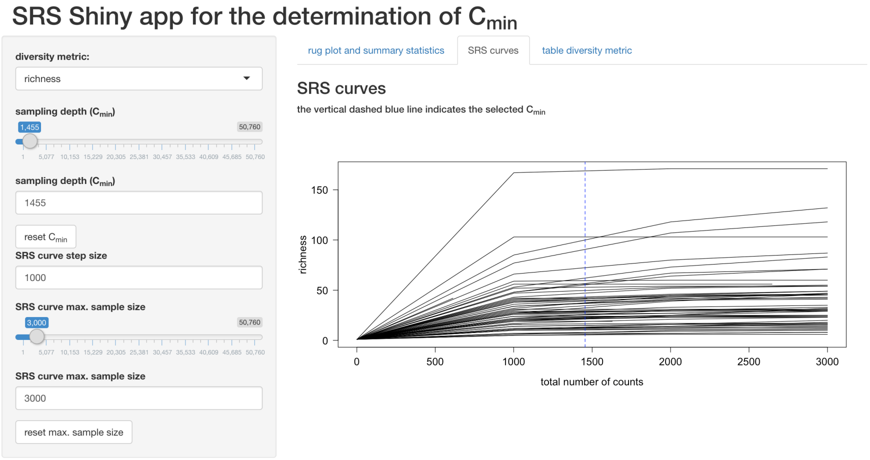
```
## Rename NAs to last known group
tax.clean <- data.frame(tax_table(phy_obj))
for (i in 1:7){ tax.clean[,i] <- as.character(tax.clean[,i])}
tax.clean[is.na(tax.clean)] <- ""

for (i in 1:nrow(tax.clean)){
  if (tax.clean[i,2] == ""){
    kingdom <- paste("Kingdom_", tax.clean[i,1], sep = "")
    tax.clean[i, 2:7] <- kingdom
  } else if (tax.clean[i,3] == ""){
    phylum <- paste("Phylum_", tax.clean[i,2], sep = "")
    tax.clean[i, 3:7] <- phylum
  } else if (tax.clean[i,4] == ""){
    class <- paste("Class_", tax.clean[i,3], sep = "")
    tax.clean[i, 4:7] <- class
  } else if (tax.clean[i,5] == ""){
    order <- paste("Order_", tax.clean[i,4], sep = "")
    tax.clean[i, 5:7] <- order
  } else if (tax.clean[i,6] == ""){
    family <- paste("Family_", tax.clean[i,5], sep = "")
    tax.clean[i, 6:7] <- family
  } else if (tax.clean[i,7] == ""){
    tax.clean$Species[i] <- paste("Genus",tax.clean$Genus[i], sep = "_")
  }
}
## import new taxonomy table
tax_table(phy_obj) <- as.matrix(tax.clean)


## save phyloseq object
saveRDS(phy_obj, "Chap2/physeq.rds")
```

Normalization with SRS

```
# phy_obj<- readRDS("Chap2/physeq.rds")

summary(sample_sums(phy_obj))
# set seed
set.seed(9242)

# pull dataframe
otu<-as.data.frame(otu_table(phy_obj))

# look as SRScurve
SRS.shiny.app(otu)
```

SRS normalization curves for species richness and shannon diversity

Figure Supplemental 1



SRS Shiny app for the determination of C_{min}

diversity metric: richness

sampling depth (C_{min}): 1,455

sampling depth (C_{min}): 1455

SRS curve step size: 1000

SRS curve max. sample size: 3,000

SRS curve max. sample size: 3000

SRS curves

the vertical dashed blue line indicates the selected C_{min}

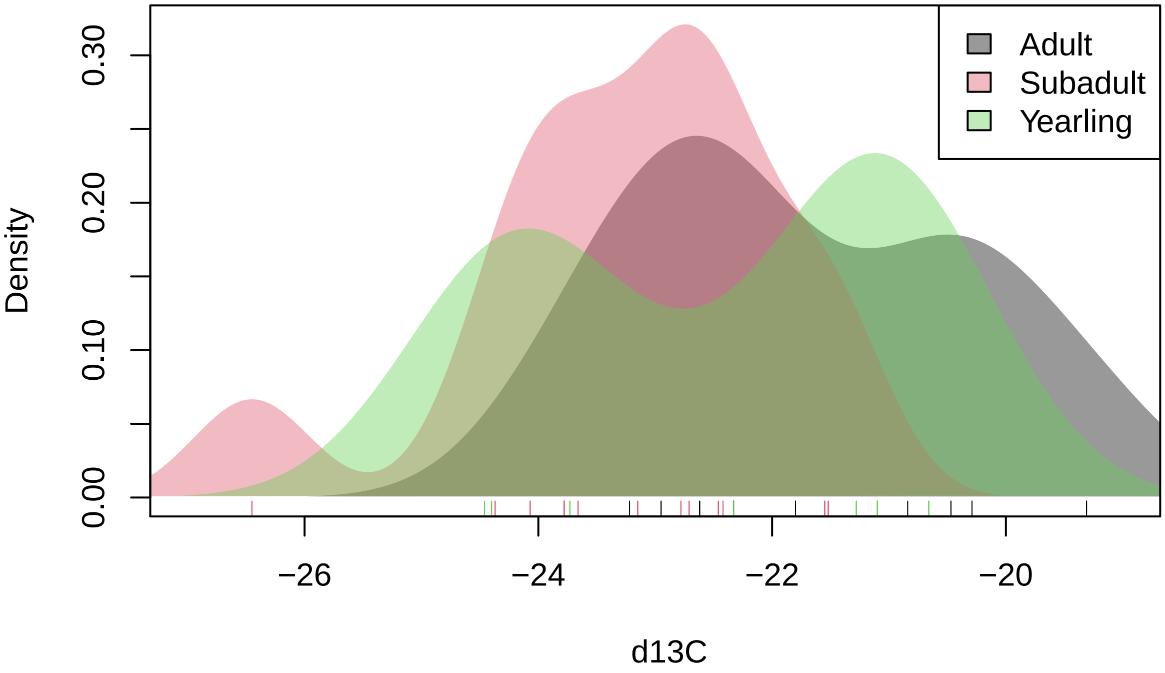
```
new_otu<-as.matrix(SRS(otu, Cmin=1455,
                       set_seed = T, seed=9242),
                   rownames=T)
# 2 sample(s) discarded due to low number
# of counts (number of counts < Cmin):  S100J, S56J

## need to get the row names back
rownames(new_otu)<-rownames(otu)

# create new phyloseq object
srs_obj<-phyloseq(otu_table(new_otu,
                            taxa_are_rows = T),
         phy_tree(phy_obj),
         tax_table(phy_obj),
         sample_data(phy_obj))

summary(sample_sums(srs_obj))
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 1455    1455    1455    1455    1455    1455
```

```
saveRDS(srs_obj, "Chap2/physeq_srs.rds")
```

Normalize

Comparison of long-term consumption of human-provisioned foods across age-classes.

```
srs_obj<-readRDS("Chap2/physeq_srs.rds")

# checking Assumption 1: normality of each group
sam.meta<-meta(srs_obj)
sam.meta$SampleID <- rownames(sam.meta)
# remove unknowns:
sam.meta2<- subset(sam.meta, AgeClass!="Unknown")

byf.hist(d13C~AgeClass, data=sam.meta2, density= T, sep=F)
```



```
# looks okay

# shapiro-wilk
byf.shapiro(d13C~AgeClass, data=sam.meta2)
```

```
##
##  Shapiro-Wilk normality tests
##
## data:  d13C by AgeClass
##
##                 W p-value
## Adult    0.9285 0.18237
## Subadult 0.9018 0.02350 *
## Yearling 0.8222 0.01695 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# adults are fine but yearlings and subadults are not..

#Assumption 2: Levene test
leveneTest(d13C~AgeClass, data=sam.meta2)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  0.9704 0.3858
##       51
```

```
# not sig.

# will go with Kruskal-wallis.

kruskal.test(d13C~AgeClass, data=sam.meta2)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  d13C by AgeClass
## Kruskal-Wallis chi-squared = 8.4698, df = 2, p-value = 0.01448
```

```
# post-hoc
posthoc.kruskal.dunn.test(d13C~as.factor(AgeClass), data=sam.meta2, p.adjust="bonf")
```

```
## Warning in posthoc.kruskal.dunn.test.default(c(-22.78, -22.78, -23.15, -23.15, :
## Ties are present. z-quantiles were corrected for ties.
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  d13C by as.factor(AgeClass)
##
##          Adult Subadult
## Subadult 0.011 -
## Yearling 0.777 0.517
##
## P value adjustment method: bonferroni
```

```
# subadult and Adult
```

Population level community composition

```
srs_rel<- microbiome::transform(srs_obj, "compositional")

comp <- tax_glom(srs_rel, taxrank = "Genus")

comp_melt<- psmelt(comp)

p_abund<-summarySE(comp_melt, measurevar = "Abundance", groupvars =c("Genus"))
```

```
##remove 0 abundance
p_abund$Abundance[p_abund$Abundance==0] <- NA
p_abund<-p_abund[complete.cases(p_abund$Abundance),]
p_abund<- p_abund %>%
  mutate_if(is.numeric, round, digits = 5)


## genus
p_abund$Genus <- as.character(p_abund$Genus)

#simple way to rename phyla with < 1% abundance
p_abund$Genus[p_abund$Abundance < 0.01] <- "Minor"
unique(p_abund$Genus)

abundance<-subset(p_abund, Genus!="Minor")
merged_abund<-merge(abundance, comp_melt, by="Genus")
merged_abund <- merged_abund[!duplicated(merged_abund$Genus),]
write.csv(merged_abund, "Chap2/abundance.csv")

# Phylum
comp <- tax_glom(srs_rel, taxrank = "Phylum")

comp_melt2<- psmelt(comp)

p_abund<-summarySE(comp_melt2, measurevar = "Abundance", groupvars =c("Phylum"))

##remove 0 abundance
p_abund$Abundance[p_abund$Abundance==0] <- NA
p_abund<-p_abund[complete.cases(p_abund$Abundance),]
p_abund<- p_abund %>%
  mutate_if(is.numeric, round, digits = 5)


## genus
p_abund$Phylum <- as.character(p_abund$Phylum)

#simple way to rename phyla with < 1% abundance
p_abund$Phylum[p_abund$Abundance < 0.01] <- "Minor"
unique(p_abund$Phylum)

abundance<-subset(p_abund, Phylum!="Minor")
abundance
```

Alpha Diversity calculations

Indices calculated:

- *Observed ASVs*
- *Faith's Phylogenetic diversity (PD)*

Observed ASVs & Faith's PD

PD function in picante package calculates both Faith's PD and observed ASVs

```
## pull ASV table
srs_obj.asvtab <- as.data.frame(srs_obj@otu_table)


## pull tree
srs_obj.tree <- srs_obj@phy_tree

## We first need to check if the tree is rooted or not
srs_obj@phy_tree
```

```
##
## Phylogenetic tree with 1605 tips and 1602 internal nodes.
##
## Tip labels:
##   6c3b47bbac4c7af75368b9aa77620a3a, e9351dc418b0f459ed95e8a2e8a62e6b, e705fdb7baa2313231f39c7b006fe37
## Node labels:
##   root, , 0.666, 0.973, 0.761, 0.862, ...
##
## Rooted; includes branch lengths.
```

```
###rooted so we are good to go

## Getting the data ready
div_pd <- pd(t(srs_obj.asvtab), srs_obj.tree,include.root=T)

## Add the rownames to diversity table
div_pd$SampleID <- rownames(div_pd)
```

Merge all of the alphas into one dataset Only keeping samples from bears with both jejunum and colon= 54

```
alpha_table <- merge(div_pd,sam.meta, by = "SampleID", all=T)
# drop samples with only one sample

 sub_counts <- alpha_table %>%
    group_by(Subject) %>%
    tally


# get names of the species with counts >= 10
new_alpha <-  sub_counts %>%
    filter(n ==2) %>%
    select(Subject) %>%
  merge(alpha_table, by="Subject")
#54 samples
```

Statisical analysis of alpha diversity

```
null_SR<-brm(SR~1+(1|Subject),
         family=negbinomial(),
         chains=4,
         iter=20000,
         warmup=2000,
         seed = 20191125,
         thin=5,
         data=new_alpha,save_all_pars = T)


modelGIT_SR<-brm(SR~d13C*GIT+(1|Subject),
         family=negbinomial(),
         chains=4,
         iter=20000,
         warmup=2000,
         seed = 20191125,
         thin=5,
         data=new_alpha,save_all_pars = T)

modelAge_SR<-brm(SR~d13C*AgeClass+(1|Subject),
         family=negbinomial(),
         chains=4,
         iter=20000,
         warmup=2000,
         seed = 20191125,
         thin=5,
         data=new_alpha, save_all_pars = T, control=list(max_treedepth=15))

model13_SR<-brm(SR~d13C+(1|Subject),
         family=negbinomial(),
         chains=4,
         iter=20000,
         warmup=2000,
         seed = 20191125,
         thin=5,
         data=new_alpha, save_all_pars = T)


SRloo_null<-loo::loo(null_SR, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
               save_psis = TRUE)

SRloo_GIT<-loo::loo(modelGIT_SR, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
               save_psis = TRUE)

SRloo_Age<-loo::loo(modelAge_SR, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
               save_psis = TRUE)

SRloo13<-loo::loo(model13_SR, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
               save_psis = TRUE)

SR_compare<-loo_compare(SRloo_null,SRloo_GIT, SRloo_Age,SRloo13)
saveRDS(SR_compare, "Chap2/SR_loo.rds")
saveRDS(model13_SR, "Chap2/SR_mod.rds")
# model without interaction is best.
```

Observed ASVs Bayesian Multilevel Models

```
##             elpd_diff se_diff elpd_loo se_elpd_loo p_loo  se_p_loo looic
## model13_SR     0.0       0.0  -238.5      6.9         9.1    1.8    476.9
## modelGIT_SR   -1.6       1.4  -240.0      6.8        10.6    1.7    480.0
## modelAge_SR   -2.9       1.7  -241.3      6.6        12.6    1.7    482.7
## null_SR       -3.6       2.5  -242.0      8.4        13.7    3.3    484.0
##             se_looic
## model13_SR    13.8
## modelGIT_SR   13.5
## modelAge_SR   13.2
## null_SR       16.9
```

```
## log PD
null_PD<-brm(log(PD)~1+(1|Subject),
         gaussian(),
         chains=4,
         iter=20000,
         warmup=2000,
         thin=5,
         seed = 20191125,
         data=new_alpha, save_all_pars = T)

modelAge_PD<-brm(log(PD)~d13C*AgeClass+(1|Subject),
         gaussian(),
         chains=4,
         iter=20000,
         warmup=2000,
         thin=5,
         seed = 20191125,
         data=new_alpha, save_all_pars = T)

modelGIT_PD<-brm(log(PD)~d13C*GIT+(1|Subject),
         gaussian(),
         chains=4,
         iter=20000,
         warmup=2000,
         thin=5,
         seed = 20191125,
         data=new_alpha, save_all_pars = T)

model13_PD<-brm(log(PD)~d13C+(1|Subject),
         gaussian(),
         chains=4,
         iter=20000,
         warmup=2000,
         thin=5,
         seed = 20191125,
         data=new_alpha, save_all_pars = T)

PDloon_null<-loo::loo(null_PD, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
                save_psis = TRUE)

PDloo_GIT<-loo::loo(modelGIT_PD, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
                save_psis = TRUE)

PDlooAge<-loo::loo(modelAge_PD, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
                save_psis = TRUE)

PDloo13<-loo::loo(model13_PD, k_threshold=.7,
                moment_match = T,
                reloo=T,seed=TRUE,
                save_psis = TRUE)


PD_compare<-loo_compare(PDloon_null,PDloo_GIT, PDlooAge,PDloo13)
saveRDS(PD_compare, "Chap2/PD_loo.rds")
saveRDS(model13_PD, "Chap2/PD_mod.rds")
# model without an interaction
```

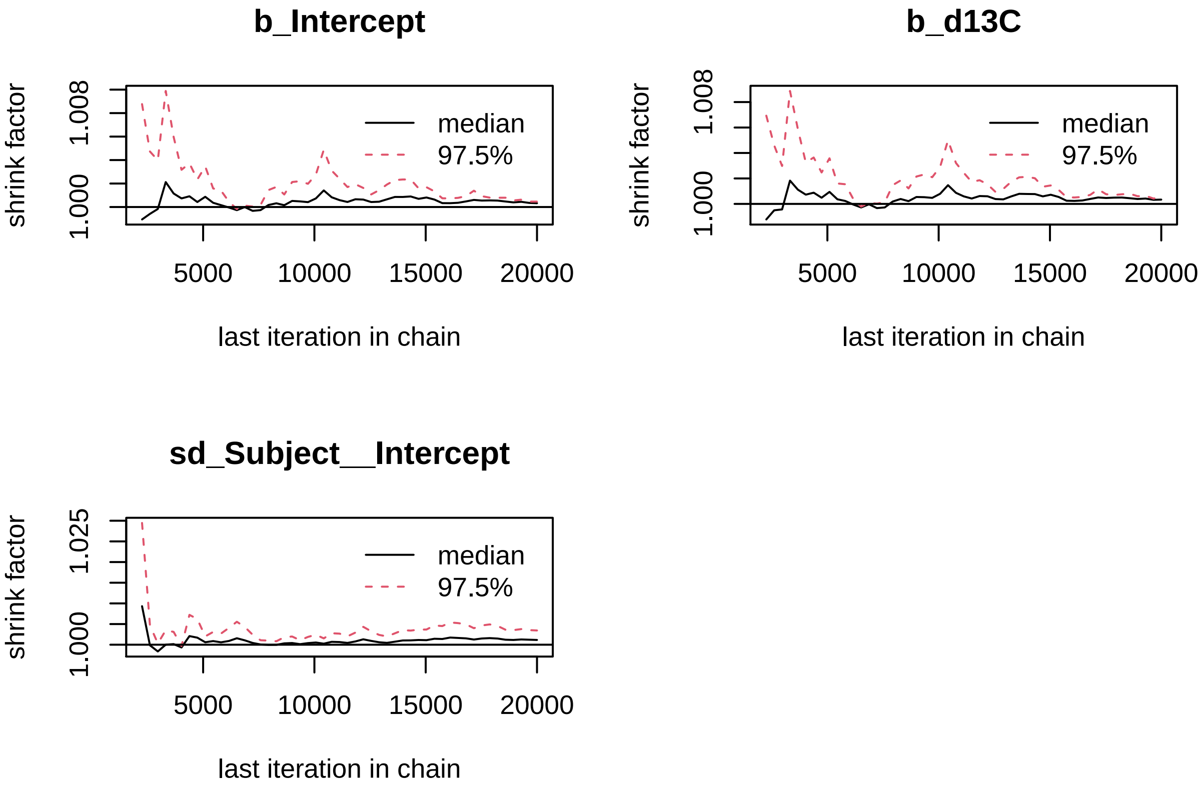
```
##             elpd_diff se_diff elpd_loo se_elpd_loo p_loo se_p_loo looic
## model13_PD    0.0       0.0   -41.5      4.8         9.5   1.5     83.0
## modelGIT_PD  -0.2       2.4   -41.7      4.6        12.1   1.9     83.4
## modelAge_PD  -2.5       2.0   -44.0      4.6        11.8   1.4     88.0
## null_PD      -2.5       2.4   -44.0      5.5        11.1   1.8     88.1
##             se_looic
## model13_PD    9.7
## modelGIT_PD   9.3
## modelAge_PD   9.3
## null_PD      10.9
```

```
# To obtain the Gelman and Rubin diagnostic use for top models:
model_SR<-readRDS("Chap2/SR_mod.rds")
# SR
modelposterior <- brms::as.mcmc(model_SR) # with the as.mcmc() command we can
# use all the CODA package convergence statistics and plotting options
gelman.diag(modelposterior[, 1:3])
```

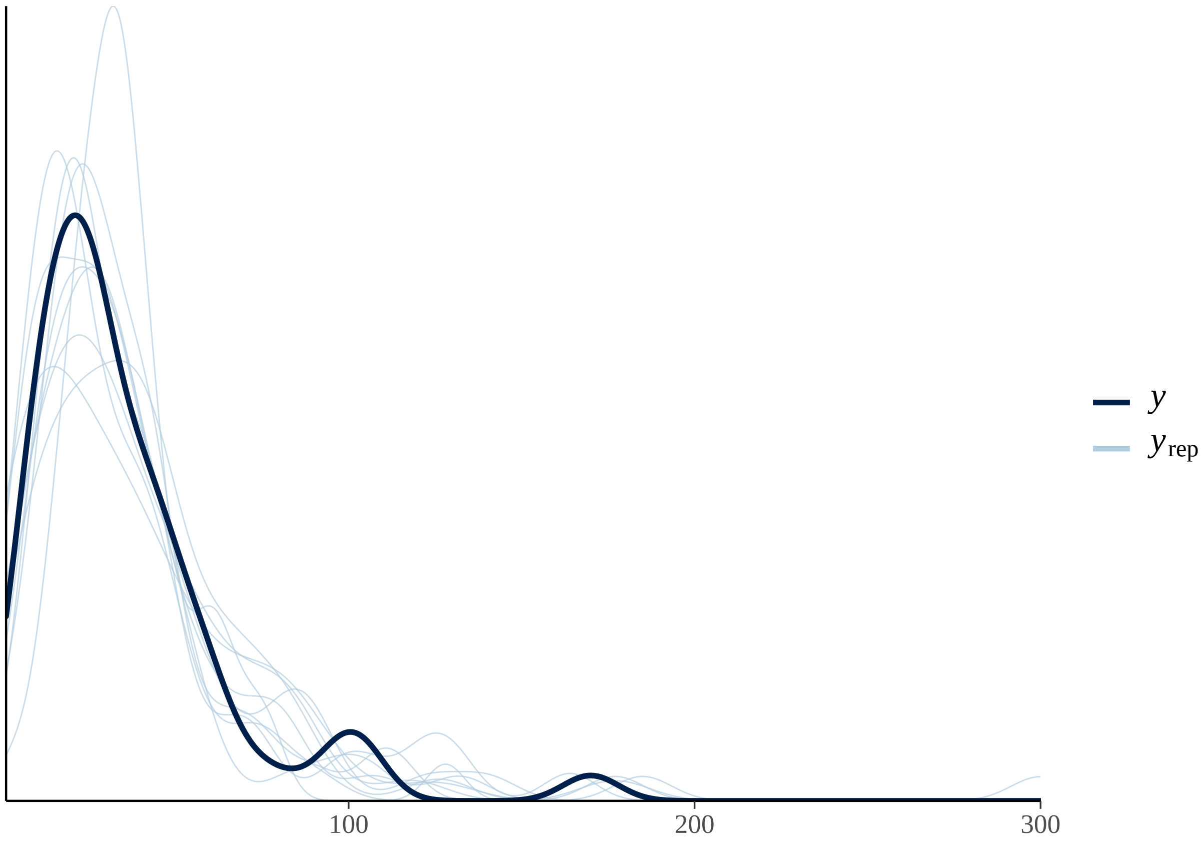
Check convergence for each model and plot

```
## Potential scale reduction factors:
##
##                       Point est. Upper C.I.
## b_Intercept                    1          1
## b_d13C                          1          1
## sd_Subject__Intercept           1          1
##
## Multivariate psrf
##
## 1
```

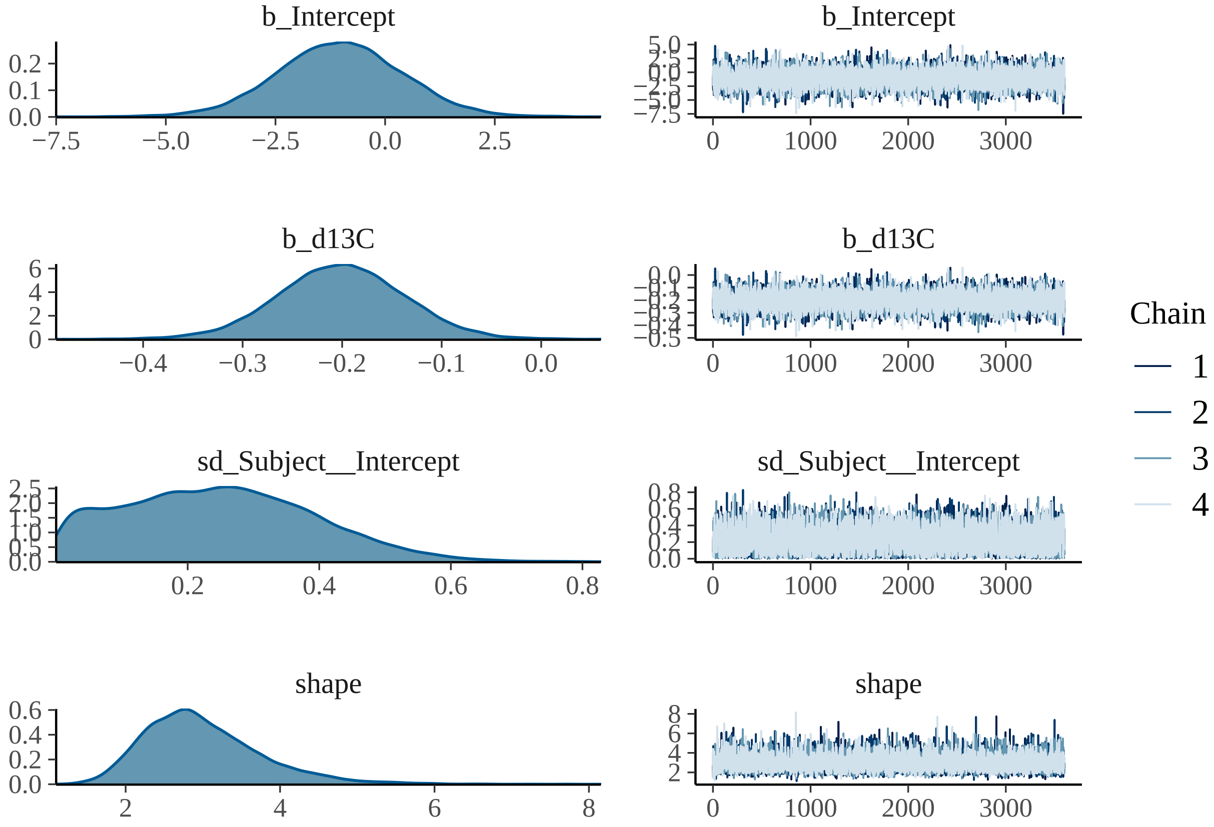
```
gelman.plot(modelposterior[, 1:3])
```



```
pp_check(model_SR)
```



```
plot(model_SR)
```



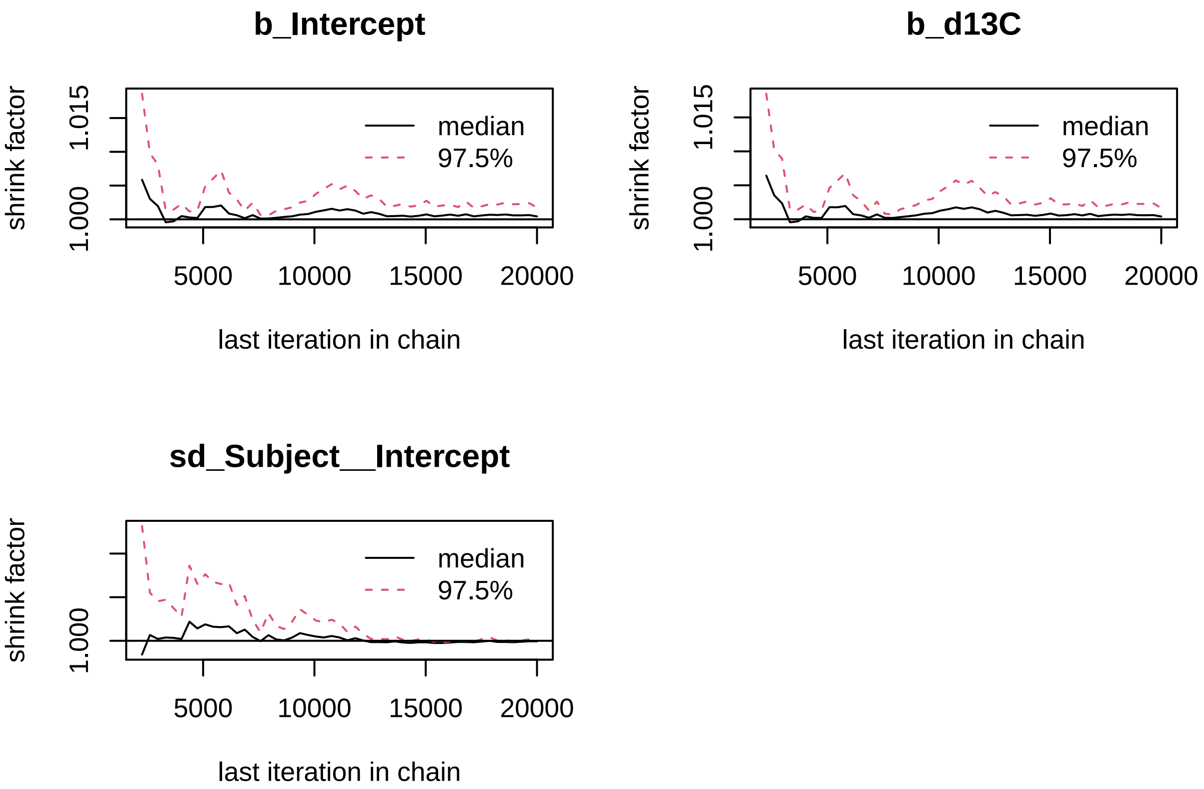
```
summary(model_SR)
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: SR ~ d13C + (1 | Subject)
##    Data: new_alpha (Number of observations: 54)
## Samples: 4 chains, each with iter = 20000; warmup = 2000; thin = 5;
##          total post-warmup samples = 14400
##
## Group-Level Effects:
## ~Subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.25      0.14     0.01     0.54 1.00    10698    13039
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -1.03      1.46    -3.91     1.86 1.00    14043    13651
## d13C         -0.20      0.06    -0.33    -0.07 1.00    14018    13477
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     3.00      0.77     1.82     4.78 1.00    12801    12823
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

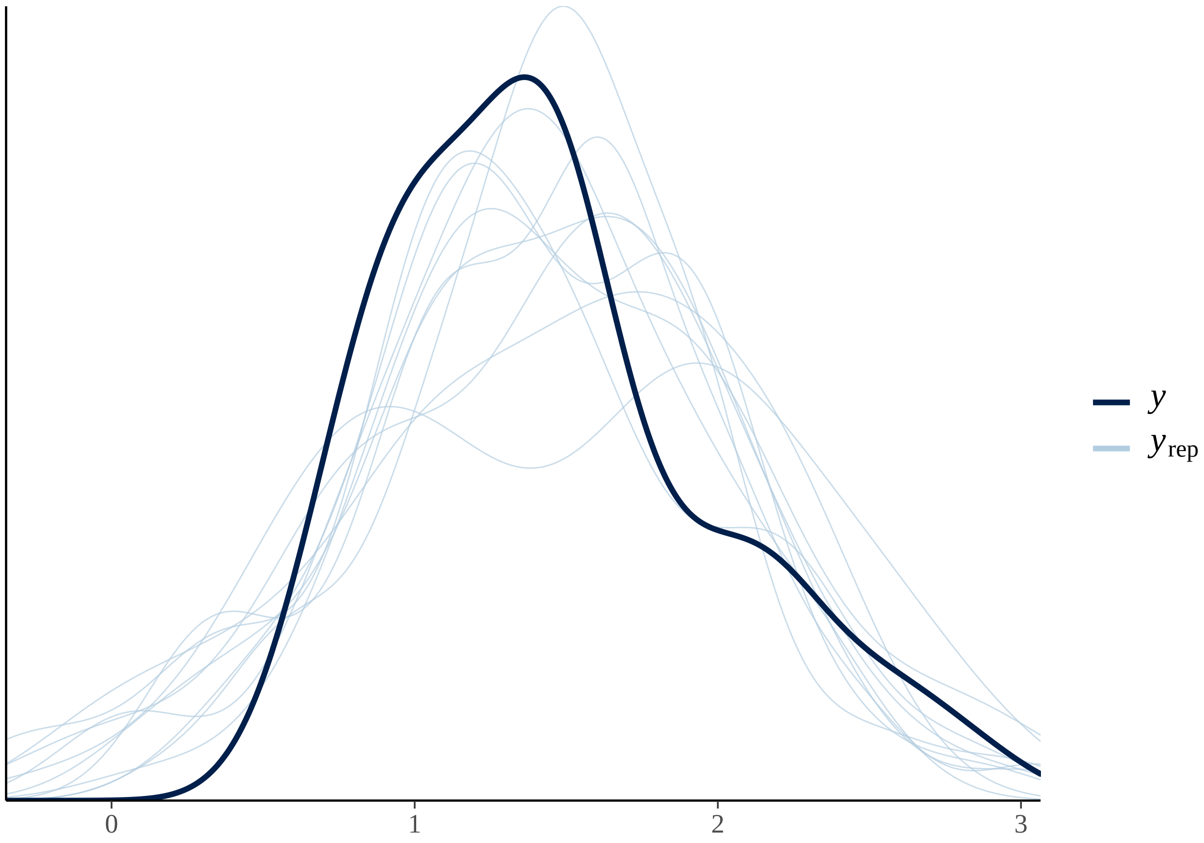
```
# PD
model_PD<-readRDS("Chap2/PD_mod.rds")
modelposterior <- brms::as.mcmc(model_PD) # with the as.mcmc() command we can
# use all the CODA package convergence statistics and plotting options
gelman.diag(modelposterior[, 1:3])
```

```
## Potential scale reduction factors:
##
##                       Point est. Upper C.I.
## b_Intercept                    1          1
## b_d13C                         1          1
## sd_Subject__Intercept          1          1
##
## Multivariate psrf
##
## 1
```

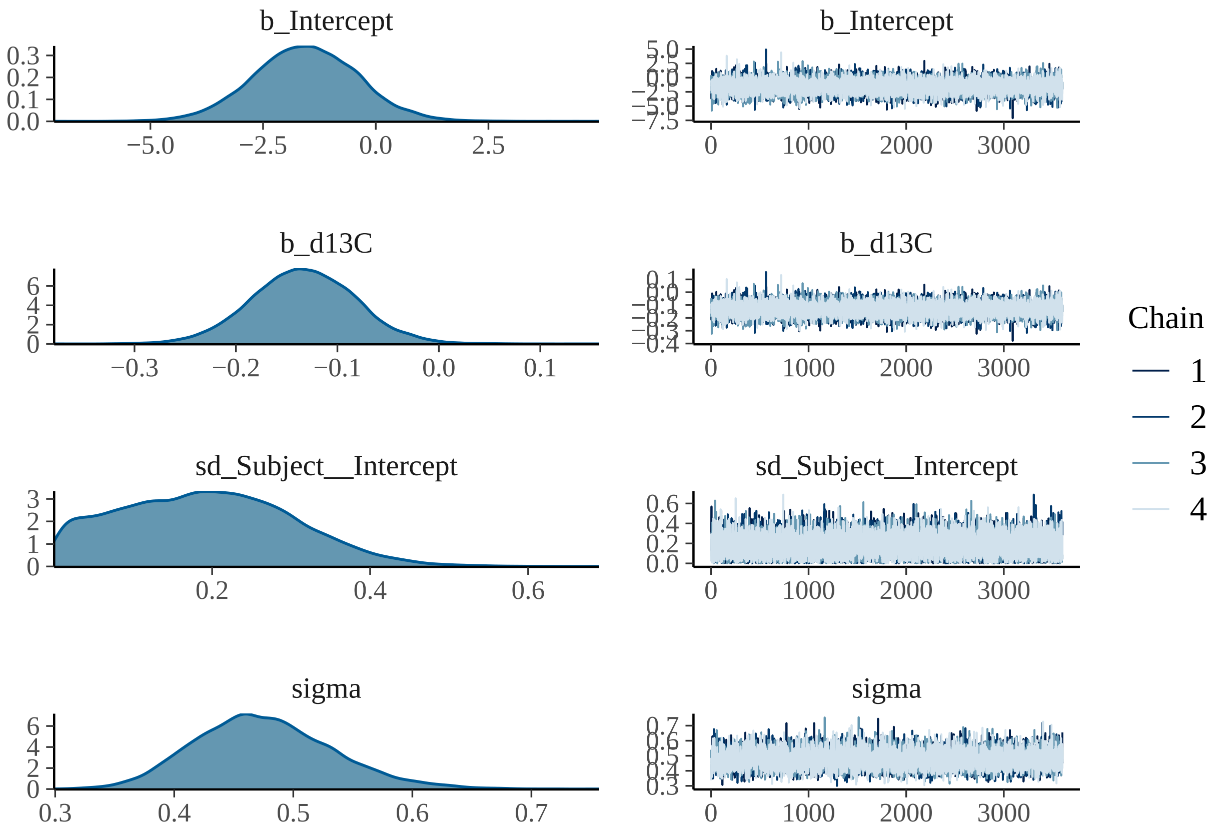
```
gelman.plot(modelposterior[, 1:3])
```



```
pp_check(model_PD)
```



```
plot(model_PD)
```



```
summary(model_PD)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: log(PD) ~ d13C + (1 | Subject)
##    Data: new_alpha (Number of observations: 54)
## Samples: 4 chains, each with iter = 20000; warmup = 2000; thin = 5;
##          total post-warmup samples = 14400
##
## Group-Level Effects:
## ~Subject (Number of levels: 27)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.19      0.11     0.01     0.41 1.00    11016    13358
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -1.55      1.16    -3.83     0.75 1.00    14505    14104
## d13C         -0.13      0.05    -0.23    -0.03 1.00    14488    14143
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.48      0.06     0.37     0.60 1.00    12012    13516
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Beta diversity statistical analysis

Weighted UniFrac Mantel's test

```
#srs_obj<- readRDS("Chap2/physeq_srs.rds")

srs_rel<- microbiome::transform(srs_obj, "compositional")

wunifrac.distC <- UniFrac(srs_rel,
                          weighted = TRUE,
                          normalized = TRUE,
                          parallel = FALSE,
                          fast = TRUE)


## mantels for carbon
external<-as.data.frame(sam.meta[,c("d13C")])

met_euc<-vegdist(external[,1], method="euc")

se.man<-mantel(wunifrac.distC,met_euc, method="spearman", permutations = 9999 )
se.man
#Mantel statistic r: 0.01952
#      Significance: 0.2752
```

Unweighted UniFrac Mantel's test

```
uunifrac.distC <- UniFrac(srs_rel,
                           weighted = FALSE,
                           normalized = TRUE,
                           parallel = FALSE,
                           fast = TRUE)




met_euc<-vegdist(external, method="euc")

se.man<-mantel(uunifrac.distC,met_euc, method="spearman", permutations = 9999)
se.man
#Mantel statistic r: 0.1043
#      Significance: 0.068
```

Plotting

Figure 1

```
## plot

library(scico)
```

```
scico(2, palette= "cork")
iso<-read.csv("Chap2/Chap2_Consumer2.csv")
p1<-ggplot(iso, aes(x=d13C, y=d15N)) +
  geom_point(color="black",shape=21,size=2.5)+
  geom_point(color="black",size=2, alpha=.8)+
  scale_color_manual(values=c("black"))+
  theme_classic()+
  theme(legend.position = "none",
    legend.title =element_text(size=10, family="Helvetica"),
    legend.title.align = 0.5,
        legend.text = element_text(size=10, family="Helvetica"),
        axis.text.x = element_text(color="black", family="Helvetica", size=10),
        axis.text.y = element_text(color="black", family="Helvetica", size=10),
        panel.background=element_blank(),panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(family="Helvetica", size=10),
        strip.text =element_text(family="Helvetica", size=10),
        axis.line = element_line(color="black"),
        strip.background = element_blank(),
        axis.title.y = element_text(family="Helvetica", size=10))+
  scale_x_continuous(breaks=c(-27,-26,-25,-24,-23,-22,-21,-20,-19),
                     limits = c(-27,-19))+
  scale_y_continuous(breaks=c(2,3,4,5,6,7), limits = c(2,7))+
  xlab(expression(paste(delta^13,"C(\u2030)")))+ylab(expression(paste(delta^15,"N(\u2030)")))

sam.meta2$AgeClass<- factor(sam.meta2$AgeClass, levels=c("Yearling", "Subadult","Adult"))

p2<-ggplot(data=sam.meta2, aes(x=AgeClass, y=d13C, fill=AgeClass))+
  geom_violin()+
 geom_boxplot(fill="white",width = 0.15, color="black",
               outlier.color = "black",
               outlier.fill="gray",outlier.shape = 21)+
  scale_fill_manual(values=c("gray","gray55", "gray25"))+
  theme_classic()+
  theme(legend.position = "none",
    legend.title =element_text(size=10, family="Helvetica"),
        legend.text = element_text(size=10, family="Helvetica"),
        axis.text.x = element_text(color="black", family="Helvetica", size=10),
        axis.text.y = element_text(color="black", family="Helvetica", size=10),
        panel.background=element_blank(),panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(family="Helvetica", size=10),
        strip.text =element_text(family="Helvetica", size=10),
        axis.line = element_line(color="black"),
        strip.background = element_blank(),
        axis.title.y = element_text(family="Helvetica", size=10))+
    scale_y_continuous(breaks=c(-27,-25,-23,-21,-19),
                       limits = c(-27,-19))+
  xlab("AgeClass")+ylab(expression(paste(delta^13,"C(\u2030)")))


fig1<-cowplot::plot_grid(p1, p2, ncol=1)
```

```
ggsave(fig1, filename="Chap2/Fig12_new.pdf", width=90, height = 90,
       encoding="MacRoman",
       units="mm",
       device="pdf")

str(pdfFonts("sans"))
```

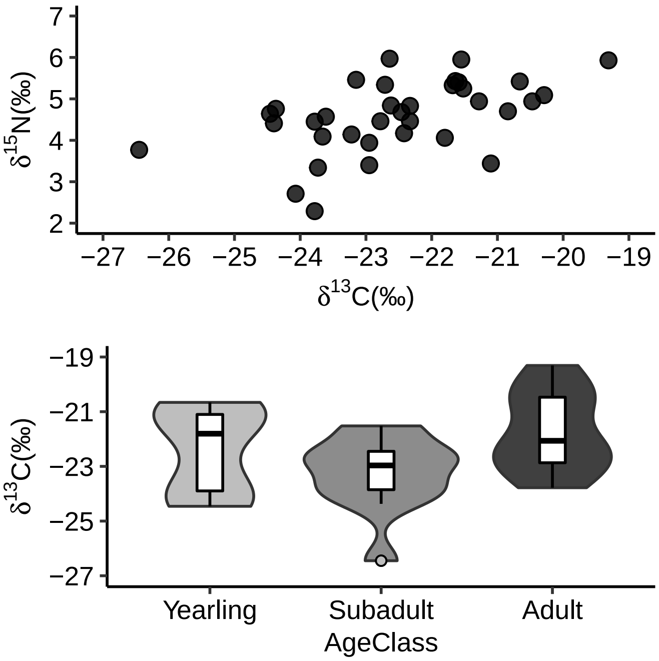
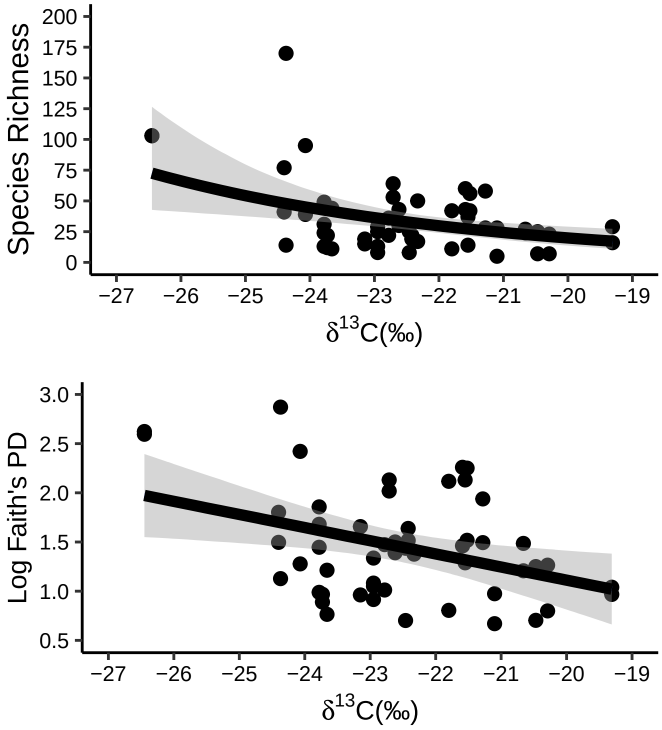


Figure 2

```
### Plot the Postier estimates
me[[1]]
me<-conditional_effects(model_PD)
p2<-plot(me, plot=F, points=T)[[1]]+
  geom_line(color="black",size=2)+
  theme_classic()+
  theme(legend.title =element_text(size=12, family="Helvetica"),
        legend.text = element_text(size=10, family="Helvetica"),
        axis.text.x = element_text(color="black", family="Helvetica", size=8),
        axis.text.y = element_text(color="black", family="Helvetica", size=8),
        panel.background=element_blank(),panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(family="Helvetica",, size=10),
        strip.text =element_text(family="Helvetica", size=10),
        axis.line = element_line(color="black"),
        strip.background = element_blank(),
        axis.title.y = element_text(family="Helvetica", size=10))+
  scale_x_continuous(breaks=c(-27,-26,-25,-24,-23,-22,
                              -21,-20,-19), limits=c(-27,-19))+
   scale_y_continuous(limits=c(0.5,3),breaks=c(0.5,1,1.5,2,2.5,3))+
        ylab("Log Faith's PD")+xlab(expression(paste(delta^13,"C(\u2030)")))
```

```
me<-conditional_effects(model_SR)
p1<-plot(me, plot=F, points=T)[[1]]+
  geom_line(color="black",size=2)+
  theme_classic()+
  theme(legend.title =element_text(size=10, family="Helvetica"),
        legend.text = element_text(size=10, family="Helvetica"),
        axis.text.x = element_text(color="black", family="Helvetica", size=8),
        axis.text.y = element_text(color="black", family="Helvetica", size=8),
        panel.background=element_blank(),panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(family="Helvetica", size=10),
        strip.text =element_text(family="Helvetica", size=10),
        axis.line = element_line(color="black"),
        strip.background = element_blank(),
        axis.title.y = element_text(family="Helvetica"))+
    scale_x_continuous(breaks=c(-27,-26,-25,-24,-23,-22,
                                -21,-20,-19),limits=c(-27,-19))+
    scale_y_continuous(breaks=c(0,25,50,75,100,125,
                                150,175,200), limit=c(0,200))+
  ylab("Species Richness")+xlab(expression(paste(delta^13,"C(\u2030)")))

fig2<-plot_grid(p1, p2, ncol=1)


ggsave(fig2, filename="Chap2/Fig2_new.pdf",width=90, height = 100,
       units="mm",
       encoding="MacRoman",
       device="pdf")
```

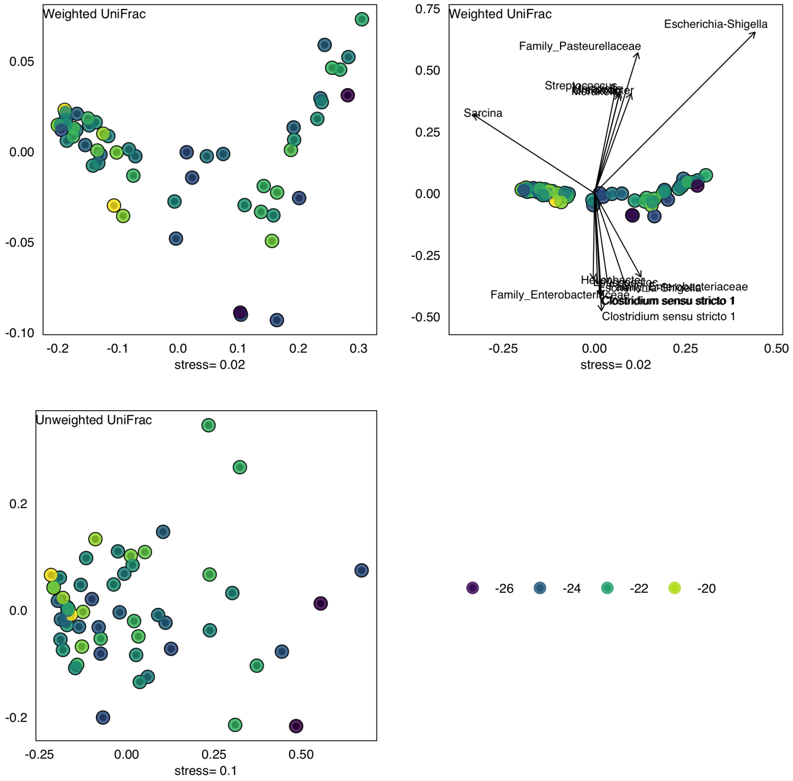


Supplemental Figures

Figure Supplemental 2

```
library(viridis)
Wunifrac.dist1<- metaMDS(wunifrac.distC ,k=3, trymax=1000, autotransform=F)
Wunifrac.dist1
# stress 0.02

p0<-phyloseq::plot_ordination(srs_rel, Wunifrac.dist1, axes=c(1,2))+
  #geom_point(color="black",shape=21,size=3.5)+
  geom_point(aes(color=as.numeric(d13C)),size=3, alpha=.8)+
  scale_color_viridis(option = "D")+
  theme(
    axis.ticks = element_blank(),
    axis.text.x = element_text(family="Helvetica",size=8, color="black"),
    axis.text.y = element_text(family="Helvetica",size=8, color="black"),
    axis.title = element_text(family="Helvetica", size=8, color="black"),
    legend.text = element_text(size=8, family="Helvetica"),
    legend.key = element_rect(fill = NA),
    axis.title.x = element_text(family="Helvetica",size=8),
    axis.title.y = element_text(size=8, family="Helvetica"),
    panel.background = element_blank(),
    legend.title = element_blank(),
    plot.title = element_text(family="Helvetica", size=8, color="black", vjust=-8),
    plot.background = element_blank(),
    panel.border = element_rect(colour = "black",
  fill=NA, size=.5))+
  guides(colour = guide_legend(nrow = 1))+
  ggtitle("Weighted UniFrac")+
  xlab("stress= 0.02")+ylab("")

p1<-phyloseq::plot_ordination(srs_rel, Wunifrac.dist1, axes=c(1,2))+
  geom_point(color="black",shape=21,size=3.5)+
  geom_point(aes(color=d13C),size=3, alpha=.8)+
  scale_color_viridis(option = "D")+
  theme(
    legend.position="none",
    axis.ticks = element_blank(),
    axis.text.x = element_text(family="Helvetica",size=8, color="black"),
    axis.text.y = element_text(family="Helvetica",size=8, color="black"),
    axis.title = element_text(family="Helvetica", size=8, color="black"),
    legend.text = element_text(size=8, family="Helvetica"),
    legend.key = element_rect(fill = NA),
    axis.title.x = element_text(family="Helvetica",size=8),
    axis.title.y = element_text(size=8, family="Helvetica"),
    panel.background = element_blank(),
    legend.title = element_blank(),
    plot.title = element_text(family="Helvetica", size=8, color="black", vjust=-8),
    plot.background = element_blank(),
    panel.border = element_rect(colour = "black",
  fill=NA, size=.5))+
  guides(colour = guide_legend(nrow = 1))+
  ggtitle("Weighted UniFrac")+
  xlab("stress= 0.02")+ylab("")


# species loadings

## with taxa
GPfr = phyloseq::filter_taxa(srs_rel, function(x) mean(x) >= 0.001, TRUE)
otu_dat<-t(as.data.frame(otu_table(GPfr)))

ef.nmdsu <- envfit(Wunifrac.dist1, otu_dat, permutations = 999)
# Now add the environmental variables as arrows
pvals<-as.data.frame(ef.nmdsu$vectors$pvals)
pvals$lab<-rownames(pvals)
pvals
sig<-pvals[which(ef.nmdsu$vectors$pvals<=0.05),]
taxa_dat<-as.data.frame(tax_table(GPfr))


arrowmat = vegan::scores(ef.nmdsu, display = "bp")
# Add labels, make a data.frame
arrowdf <- data.frame(lab = rownames(arrowmat), arrowmat)

taxa_dat$lab<-rownames(taxa_dat)


darrow<-semi_join(arrowdf,sig, by="lab")%>%
  semi_join(taxa_dat, by="lab")


taxmerg<-taxa_dat[,c("Genus","lab")]

arrdowd<-merge(darrow, taxmerg, by="lab")
arrdowd<-arrdowd[,-1]


# Define the arrow aesthetic mapping
arrow_map = aes(xend = NMDS1, yend = NMDS2, x = 0, y = 0, shape = NULL, color = NULL)
label_map = aes(x = 1.08 * NMDS1, y = 1.08 * NMDS2, shape = NULL, color = NULL)

# Make a new graphic
arrowhead = arrow(length = unit(0.02, "npc"))

p2<-phyloseq::plot_ordination(srs_rel, Wunifrac.dist1, color="d13C", axes=c(1,2))+
  geom_point(color="black",shape=21,size=3.5)+
  geom_point(aes(color=d13C),size=3, alpha=.8)+
  geom_segment(arrow_map, size = 0.3, data = arrdowd, color = "black",
               arrow = arrowhead) +
  geom_text( vjust="inward",hjust="inward",
             label_map, size = 2.5, label=arrdowd$Genus,data = arrdowd, family="Helvetica")+
  scale_color_viridis(option = "D")+
  theme(
    legend.position="none",
    axis.ticks = element_blank(),
    axis.text.x = element_text(family="Helvetica",size=8, color="black"),
    axis.text.y = element_text(family="Helvetica",size=8, color="black"),
    axis.title = element_text(family="Helvetica", size=8, color="black"),
    legend.text = element_text(size=8, family="Helvetica"),
    legend.key = element_rect(fill = NA),
    axis.title.x = element_text(family="Helvetica",size=8),
    axis.title.y = element_text(size=8, family="Helvetica"),
    panel.background = element_blank(),
    legend.title = element_blank(),
    plot.title = element_text(family="Helvetica", size=8, color="black", vjust=-8),
    plot.background = element_blank(),
    panel.border = element_rect(colour = "black",
  fill=NA, size=.5))+
  guides(colour = guide_legend(nrow = 1))+
  ggtitle("Weighted UniFrac")+
  xlab("stress= 0.02")+ylab("")
```

```
Uunifrac.dist1<- metaMDS(uunifrac.distC ,k=3, trymax=1000, autotransform=F)
# stress 0.10

p3<-phyloseq::plot_ordination(srs_obj, Uunifrac.dist1, axes=c(1,2))+
  geom_point(color="black",shape=21,size=3.5)+
  geom_point(aes(color=d13C),size=3, alpha=.8)+
  scale_color_viridis(option = "D")+
  theme(
    legend.position="none",
    axis.ticks = element_blank(),
    axis.text.x = element_text(family="Helvetica",size=8, color="black"),
    axis.text.y = element_text(family="Helvetica",size=8, color="black"),
    axis.title = element_text(family="Helvetica", size=8, color="black"),
    legend.text = element_text(size=8, family="Helvetica"),
    legend.key = element_rect(fill = NA),
    axis.title.x = element_text(family="Helvetica",size=8),
    axis.title.y = element_text(size=8, family="Helvetica"),
    panel.background = element_blank(),
    legend.title = element_blank(),
    plot.title = element_text(family="Helvetica", size=8, color="black", vjust=-8),
    plot.background = element_blank(),
    panel.border = element_rect(colour = "black",
  fill=NA, size=.5))+
  guides(colour = guide_legend(nrow = 1))+
  ggtitle("Unweighted UniFrac")+
  xlab("stress= 0.1")+ylab("")


legendw<-get_legend(p0)
p5<-cowplot::plot_grid(p1,p2,p3,legendw)
ggsave(p5, filename="Chap2/SupFig1.png", device="png", dpi=300)
```


Weighted UniFrac
0.05
0.00
-0.05
-0.10
-0.2
-0.1
0.0
0.1
0.2
0.3
stress= 0.02
Weighted UniFrac
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.25
0.00
0.25
0.50
stress= 0.02
Escherichia-Shigella
Family_Pasteurellaceae
Streptococcus
Moraxella
Sarcina
Helicobacter
Family_Enterobacteriaceae
Escherichia-Shigella
Clostridium sensu stricto 1
Clostridium sensu stricto 1
Unweighted UniFrac
0.2
0.0
-0.2
-0.25
0.00
0.25
0.50
stress= 0.1
● -26
● -24
● -22
● -20