***Below is the R code used for Model 2: metabarcoding data set from Indiana.*** *The data used is available as a supplemental file (“Data Sheet 3.csv”). To recreate this analysis download data.*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* R Code for Ecological Modeling & Figures \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Load required R packages

# Note that this R script requires that JAGS (Just Another Gibbs Sampler)

# is installed on your computer. Please see https://mcmc-jags.sourceforge.io/

# for more detail.

library(R2jags)

# Load Indiana Data Example. Supplemental File “Data Sheet 3”

file <- "Indiana data example.csv"

df.in <- read.csv(file)

# Format data into list to pass to JAGS

# Note that a small positive constant was added to y2 below. This was

# required to enable the use of the gamma probability distribution function

# when zeros occur. Ideally this would be avoided (e.g., as in y1 and in KS example).

# Future studies may want to try using a Tweedie distribution or a zero-inflated

# gamma distribution

df.jags <- list(y1=df.in$ASV\_1,

y2=df.in$ASV\_78+1,

y3=df.in$bydv,

x1=df.in$spring\_temp-mean(df.in$spring\_temp),

n.in=length(df.in$bydv))

# Specify Bayesian model for JAGS.

# Note that 10.6 was subtracted from y1 in the linear

# predictor for p3. This is to aid in model fitting and is

# similar to "centering a covariate" in a regression model.

jags.model <- function(){

# Likelihood:

for (i in 1:n.in){

y1[i] ~ dgamma(mu1[i]/phi1,1/phi1)

mu1[i] <- exp(alpha0\_1+alpha1\_1\*x1[i])

y2[i] ~ dgamma(mu2[i]/phi2,1/phi2)

mu2[i] <- exp(alpha0\_2+alpha1\_2\*x1[i])

y3[i] ~ dbern(p3[i])

p3[i] <- ilogit(beta0+beta1\*x1[i]+beta2\*(log(y1[i])-10.6)+beta3\*(log(y2[i])))

}

# Priors:

alpha0\_1 ~ dnorm(0,1/100)

alpha1\_1 ~ dnorm(0,1/100)

alpha0\_2 ~ dnorm(0,1/100)

alpha1\_2 ~ dnorm(0,1/100)

phi1 ~ dunif(0,3\*10^6)

phi2 ~ dunif(0,3\*10^6)

beta0 ~ dnorm(0,1/2.25)

beta1 ~ dnorm(0,1/2.25)

beta2 ~ dnorm(0,1/2.25)

beta3 ~ dnorm(0,1/2.25)

}

# Decide what random variables (parameters) to save as output

params <- c("alpha0\_1","alpha1\_1","alpha0\_2","alpha1\_2","beta0","beta1","beta2","beta3","phi1","phi2")

# Provide initial value for MCMC algorithm

init.values <- function(){

list(alpha0\_1 = sample(c(10,11,12),1),alpha1\_1 = sample(c(0,0.1,0.2),1),

alpha0\_2 = sample(c(-5,-4,-3),1),alpha1\_2 = sample(c(-1,0,1),1),

beta0 = sample(c(-1.5,-1,0.5),1), beta1 = sample(c(-1,0,1),1),beta2 = sample(c(-1,0,1),1),beta3 = sample(c(-1,0,1),1),

phi1 = sample(c(10^6/2,10^6,1.5\*10^6),1),phi2 = sample(c(10^6/2,10^6,1.5\*10^6),1))

}

# Fit Bayesian model to data using a MCMC algorithm.

# Note this will take ~1-5 min depending on your computer

ptm <- proc.time()

m.fit <- jags.parallel(data = df.jags, inits = init.values,parameters.to.save = params, model.file = jags.model,

n.chains = 4, n.iter = 3\*10^5, n.burnin = 10^3, n.thin = 100, DIC = F,jags.seed = 1229)

proc.time() - ptm

# Examine trace plots to assess MCMC algorithm

traceplot(m.fit, mfrow = c(2, 2), ask = F)

# Examine posterior distribution for parameters of interest

# Note that burn.in is set to zero because it was specified

# in the jags.parallel() function during model fitting.

# A longer burn.in may be needed depending on the data set and

# model.

burn.in <- 0

mcmc.samples <- rbind(as.mcmc(m.fit)[[1]][-c(1:burn.in),],as.mcmc(m.fit)[[2]][-c(1:burn.in),],as.mcmc(m.fit)[[3]][-c(1:burn.in),],as.mcmc(m.fit)[[4]][-c(1:burn.in),])

hist(mcmc.samples[,1],xlab=expression(alpha[0]\*"|"\*bold(y1)),ylab=expression("["\*alpha[0]\*"|"\*bold(y1)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,3],xlab=expression(alpha[1]\*"|"\*bold(y1)),ylab=expression("["\*alpha[1]\*"|"\*bold(y1)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,2],xlab=expression(alpha[0]\*"|"\*bold(y2)),ylab=expression("["\*alpha[0]\*"|"\*bold(y1)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,4],xlab=expression(alpha[1]\*"|"\*bold(y2)),ylab=expression("["\*alpha[1]\*"|"\*bold(y1)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,5],xlab=expression(beta[0]\*"|"\*bold(y1)),ylab=expression("["\*beta[0]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,6],xlab=expression(beta[1]\*"|"\*bold(y1)),ylab=expression("["\*beta[1]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,7],xlab=expression(beta[2]\*"|"\*bold(y1)),ylab=expression("["\*beta[2]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,8],xlab=expression(beta[2]\*"|"\*bold(y1)),ylab=expression("["\*beta[2]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

# The code below makes Fig. 3 as shown in the manuscript.

# Note that the code below requires a solid knowledge of

# composition sampling and the posterior predictive distribution

# see Ch. 12 in Hooten and Hefley (2019) Bringing Bayesian Models to

# Life. CRC Press.

par(mfrow=c(2,3))

# Plot showing relationship between temp (x1) and buchnera relative abundance

x1 <- seq(-1.5,8,by=0.1)

x1.bar <- mean(df.in$spring\_temp)

K <- dim(mcmc.samples)[1]

E.y1 <- matrix(,K,length(x1))

for(k in 1:K){

alpha0 <- mcmc.samples[k,1]

alpha1 <- mcmc.samples[k,3]

E.y1[k,] <- exp(alpha0+alpha1\*x1)

}

E.Y <- apply(E.y1,2,mean)

u.CI <- apply(E.y1,2,quantile,prob=0.975)

l.CI <- apply(E.y1,2,quantile,prob=0.025)

plot(x1+x1.bar,E.Y,typ="l",lwd=3,ylim=c(0,0.8\*10^5),xlab="Spring temperature (°C)",ylab="Buchnera relative abundance")

polygon(c(x1+x1.bar,rev(x1+x1.bar)),c(u.CI,rev(l.CI)),

col=rgb(0.5,0.5,0.5,0.5),border=rgb(0.5,0.5,0.5,0.5))

text(7.5, 0.78\*10^5, "(a)", cex = 1.2)

# Plot showing relationship between temp (x1) and Serratia relative abundance

x1 <- seq(-1.5,8,by=0.1)

x1.bar <- mean(df.in$spring\_temp)

K <- dim(mcmc.samples)[1]

E.y2 <- matrix(,K,length(x1))

for(k in 1:K){

alpha0 <- mcmc.samples[k,2]

alpha1 <- mcmc.samples[k,4]

E.y2[k,] <- exp(alpha0+alpha1\*x1)

}

E.Y <- apply(E.y2,2,mean)

u.CI <- apply(E.y2,2,quantile,prob=0.975)

l.CI <- apply(E.y2,2,quantile,prob=0.025)

plot(x1+x1.bar,E.Y,typ="l",lwd=3,ylim=c(0,100),xlab="Spring temperature (°C)",ylab="Serratia relative abundance")

polygon(c(x1+x1.bar,rev(x1+x1.bar)),c(u.CI,rev(l.CI)),

col=rgb(0.5,0.5,0.5,0.5),border=rgb(0.5,0.5,0.5,0.5))

text(7.5, 98, "(b)", cex = 1.2)

plot(0, xaxt = 'n', yaxt = 'n', bty = 'n', pch = '', ylab = '', xlab = '')

# Plot showing relationship between temp (x1) and probability of BYDV infection

x1 <- seq(-1.5,8,by=0.1)

y1 <- mean(df.jags$y1)

y2 <- mean(df.jags$y2)

x1.bar <- mean(df.in$spring\_temp)

K <- dim(mcmc.samples)[1]

E.y3 <- matrix(,K,length(x1))

for(k in 1:K){

beta0 <- mcmc.samples[k,5]

beta1 <- mcmc.samples[k,6]

beta2 <- mcmc.samples[k,7]

beta3 <- mcmc.samples[k,8]

E.y3[k,] <- 1/(1+exp(-(beta0+beta1\*x1+beta2\*(log(y1)-10.6)+beta3\*(log(y2)))))

}

E.Y <- apply(E.y3,2,mean)

u.CI <- apply(E.y3,2,quantile,prob=0.975)

l.CI <- apply(E.y3,2,quantile,prob=0.025)

plot(x1+x1.bar,E.Y,typ="l",lwd=3,ylim=c(0,1),xlab="Spring temperature (°C)",ylab="Proability of BYDV infection")

polygon(c(x1+x1.bar,rev(x1+x1.bar)),c(u.CI,rev(l.CI)),

col=rgb(0.5,0.5,0.5,0.5),border=rgb(0.5,0.5,0.5,0.5))

text(7.5, 0.97, "(c)", cex = 1.2)

# Plot showing relationship between buchnera relative abundance and probability of BYDV infection

x1 <- 0

y1 <- seq(194,83248,by=1000)

y2 <- mean(df.jags$y2)

x1.bar <- mean(df.in$spring\_temp)

K <- dim(mcmc.samples)[1]

E.y3 <- matrix(,K,length(y1))

for(k in 1:K){

beta0 <- mcmc.samples[k,5]

beta1 <- mcmc.samples[k,6]

beta2 <- mcmc.samples[k,7]

beta3 <- mcmc.samples[k,8]

E.y3[k,] <- 1/(1+exp(-(beta0+beta1\*x1+beta2\*(log(y1)-10.6)+beta3\*(log(y2)))))

}

E.Y <- apply(E.y3,2,mean)

u.CI <- apply(E.y3,2,quantile,prob=0.975)

l.CI <- apply(E.y3,2,quantile,prob=0.025)

plot(y1,E.Y,typ="l",lwd=3,ylim=c(0,1),xlab="Buchnera relative abundance",ylab="Proability of BYDV infection")

polygon(c(y1,rev(y1)),c(u.CI,rev(l.CI)),

col=rgb(0.5,0.5,0.5,0.5),border=rgb(0.5,0.5,0.5,0.5))

text(7000, 0.97, "(d)", cex = 1.2)

# Plot showing relationship between Serratia relative abundance and probability of BYDV infection

x1 <- 0

y1 <- mean(df.jags$y1)

y2 <- seq(1,197,by=0.1)

x1.bar <- mean(df.in$spring\_temp)

K <- dim(mcmc.samples)[1]

E.y3 <- matrix(,K,length(y2))

for(k in 1:K){

beta0 <- mcmc.samples[k,5]

beta1 <- mcmc.samples[k,6]

beta2 <- mcmc.samples[k,7]

beta3 <- mcmc.samples[k,8]

E.y3[k,] <- 1/(1+exp(-(beta0+beta1\*x1+beta2\*(log(y1)-10.6)+beta3\*(log(y2)))))

}

E.Y <- apply(E.y3,2,mean)

u.CI <- apply(E.y3,2,quantile,prob=0.975)

l.CI <- apply(E.y3,2,quantile,prob=0.025)

plot(y2,E.Y,typ="l",lwd=3,ylim=c(0,1),xlab="Serratia relative abundance",ylab="Proability of BYDV infection")

polygon(c(y2,rev(y2)),c(u.CI,rev(l.CI)),

col=rgb(0.5,0.5,0.5,0.5),border=rgb(0.5,0.5,0.5,0.5))

text(20, 0.97, "(e)", cex = 1.2)