***Below is the R code used for Model 1:* *qPCR data set from Kansas.*** *To recreate this analysis download* ***two data files*** *from supplemental files:*

*“Data Sheet 1.csv” = all-data-with-covariates.csv “Data Sheet 2.csv”= buchnera\_data.csv*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 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 data from Enders et al. (2019). Data from: Spatio-temporal distribution

# and environmental drivers of Barley yellow dwarf virus and vector abundance

# in Kansas. Dryad. https://doi.org/10.5061/dryad.mk69rg6

# This will require you to open the compressed folder and find the file

# "all-data-with-covariates.csv"

file <- "all-data-with-covariates.csv"

df.phyto <- read.csv(file)

df.phyto <- df.phyto[which(df.phyto$year==2015),]

df.phyto$ega.bydv <- ifelse(df.phyto$EGA.BYDV=="Present",1,df.phyto$EGA.BYDV)

df.phyto$ega.bydv <- ifelse(df.phyto$EGA.BYDV=="Absent",0,df.phyto$ega.bydv)

df.phyto$ega.bydv <- ifelse(df.phyto$EGA.BYDV=="Unsampled",NA,df.phyto$ega.bydv)

df.phyto$ega.bydv <- as.numeric(df.phyto$ega.bydv)

df.phyto$n.ega.bydv <- df.phyto$EGA.totaltested

# Load Buchnera data

file <- "buchnera\_data.csv"

df.mb <- read.csv(file)

df.mb$RAc <- df.mb$buchnera.relative.abundance

df.mb$bydv <- ifelse(df.mb$BYDV=="VIR",1,0)

# Format data into list to pass to JAGS

df.temp <- df.phyto

df.temp <- df.temp[which(df.temp$year==2015),]

df.temp <- df.temp[which(df.temp$lat %in% df.mb$lat & df.temp$long %in% df.mb$long),]

df.temp <- cbind(df.mb[which(df.mb$lat %in% df.temp$lat[1:6] & df.mb$long %in% df.temp$long[1:6]),],df.temp[rep(1:6,each=10),c(13:20)])

df.temp <- df.temp[-c(3,5,44),]

df.mb <- df.temp

df.jags <- list(y1=df.mb$buchnera.relative.abundance,

 y2=df.mb$bydv,

 y3=rep(NA,length(df.phyto$EGA)),

 y4=df.phyto$EGA,

 y5=df.phyto$ega.bydv,

 y5.n=df.phyto$n.ega.bydv,

 x1=df.mb$tmean.spring-mean(df.mb$tmean.spring),

 x2=df.phyto$tmean.spring-mean(df.mb$tmean.spring),

 n.mb=length(df.mb$buchnera.relative.abundance),

 n.phyto=length(df.phyto$EGA))

# Specify Bayesian model for JAGS.

# Note that 12.55 was subtracted from y1 and y3 in the linear

# predictor for p2, mu4 and p5. 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.mb){

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

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

 y2[i] ~ dbern(p2[i])

 p2[i] <- ilogit(beta0+beta1\*x1[i]+beta2\*(log(y1[i])-12.55))

 }

 for (i in 1:n.phyto){

 y3[i] ~ dgamma(mu3[i]/phi1,1/phi1)

 mu3[i] <- exp(alpha0+alpha1\*x2[i])

 y4[i] ~ dnegbin(1/phi4,mu4[i]/((1-1/phi4)\*phi4))

 mu4[i] <- exp(gamma0+gamma1\*x2[i]+gamma2\*(log(y3[i])-12.55))

 y5[i] ~ dbern(1-(1-p5[i])^y5.n[i])

 p5[i] <- ilogit(beta0+beta1\*x2[i]+beta2\*(log(y3[i])-12.55))

 }

 # Priors:

 alpha0 ~ dnorm(0,1/100)

 alpha1 ~ dunif(0,10^6)

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

 beta0 ~ dnorm(0,1/2.25)

 beta1 ~ dnorm(0,1/2.25)

 beta2 ~ dnorm(0,1/2.25)

 gamma0 ~ dnorm(0,1/100)

 gamma1 ~ dunif(0,10^6)

 gamma2 ~ dunif(0,10^6)

 phi4 ~ dunif(1,3\*10^6)

}

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

params <- c("alpha0","alpha1","beta0", "beta1","beta2","gamma0", "gamma1","gamma2","phi1","phi4")

# Provide initial value for MCMC algorithm

init.values <- function(){

 list(alpha0 = sample(c(12,13,14),1),alpha1 = sample(c(0,1,2),1),

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

 gamma0 = sample(c(3,4,5),1), gamma1 = sample(c(0,1,2),1), gamma2 = sample(c(0,1,2),1),

 phi1 = sample(c(10^6/2,10^6,1.5\*10^6),1),phi4 = sample(c(10,100,200),1))

}

# Fit Bayesian model to data using a MCMC algorithm

# Note this will take several hours 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\*10^6, n.burnin = 10^5, n.thin = 1000, 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.

burn.in <- 1900

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

hist(mcmc.samples[,3],xlab=expression(beta[0]\*"|"\*bold(y1)),ylab=expression("["\*beta[0]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

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

hist(mcmc.samples[,5],xlab=expression(beta[2]\*"|"\*bold(y1)),ylab=expression("["\*beta[2]\*"|"\*bold(y2)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,6],xlab=expression(gamma[0]\*"|"\*bold(y1)),ylab=expression("["\*gamma[0]\*"|"\*bold(y4)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,7],xlab=expression(gamma[1]\*"|"\*bold(y1)),ylab=expression("["\*gamma[1]\*"|"\*bold(y4)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

hist(mcmc.samples[,8],xlab=expression(gamma[2]\*"|"\*bold(y1)),ylab=expression("["\*gamma[2]\*"|"\*bold(y4)\*"]"),freq=FALSE,col="grey",main="",breaks=30)

# The code below makes Fig. 2 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(3,2))

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

x1 <- seq(-1.5,1.5,by=0.01)

x1.bar <- mean(df.mb$tmean.spring)

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,2]

 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(0,1.3\*10^6),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(4.7, 1.2\*10^6, "(a)", 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(-1.5,1.5,by=0.01)

y1 <- mean(df.jags$y1)

x1.bar <- mean(df.mb$tmean.spring)

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

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

for(k in 1:K){

 beta0 <- mcmc.samples[k,3]

 beta1 <- mcmc.samples[k,4]

 beta2 <- mcmc.samples[k,5]

 E.y2[k,] <- 1/(1+exp(-(beta0+beta1\*x1+beta2\*(log(y1)-12.55))))

}

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(0,0.8),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(4.7, 0.75, "(b)", cex = 1.2)

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

x1 <- 0

y1 <- seq(40000,5\*10^6,by=1000)

x1.bar <- mean(df.mb$tmean.spring)

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

E.y2 <- matrix(,K,length(y1))

for(k in 1:K){

 beta0 <- mcmc.samples[k,3]

 beta1 <- mcmc.samples[k,4]

 beta2 <- mcmc.samples[k,5]

 E.y2[k,] <- 1/(1+exp(-(beta0+beta1\*x1+beta2\*(log(y1)-12.55))))

}

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(y1,E.Y,typ="l",lwd=3,ylim=c(0,0.8),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(250000, 0.75, "(c)", cex = 1.2)

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

x1 <- seq(-1.5,1.5,by=0.01)

y1 <- mean(df.jags$y1)

x1.bar <- mean(df.mb$tmean.spring)

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

E.y4 <- matrix(,K,length(x1))

for(k in 1:K){

 gamma0 <- mcmc.samples[k,6]

 gamma1 <- mcmc.samples[k,7]

 gamma2 <- mcmc.samples[k,8]

 E.y4[k,] <- exp(gamma0+gamma1\*x1+gamma2\*(log(y1)-12.55))

}

E.Y <- apply(E.y4,2,mean)

u.CI <- apply(E.y4,2,quantile,prob=0.975)

l.CI <- apply(E.y4,2,quantile,prob=0.025)

plot(x1+x1.bar,E.Y,typ="l",lwd=3,ylim=c(0,100),xlab="Spring temperature (°C)",ylab="S.avenae 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(4.7, 92, "(d)", cex = 1.2)

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

x1 <- 0

y1 <- seq(40000,5\*10^6,by=1000)

x1.bar <- mean(df.mb$tmean.spring)

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

E.y4 <- matrix(,K,length(y1))

for(k in 1:K){

 gamma0 <- mcmc.samples[k,6]

 gamma1 <- mcmc.samples[k,7]

 gamma2 <- mcmc.samples[k,8]

 E.y4[k,] <- exp(gamma0+gamma1\*x1+gamma2\*(log(y1)-12.55))

}

E.Y <- apply(E.y4,2,mean)

u.CI <- apply(E.y4,2,quantile,prob=0.975)

l.CI <- apply(E.y4,2,quantile,prob=0.025)

plot(y1,E.Y,typ="l",lwd=3,ylim=c(0,100),xlab="Buchnera relative abundance",ylab="S.avenae relative abundance")

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(250000, 92, "(e)", cex = 1.2)