

## Appendix S2. Hierarchical group model JAGS code

The model code below is written for program R and uses the R2jags package to run JAGS.

```
#The data file, 'D', includes columns representing the study site, species (SppCode), number of
#days during which the species was photographed  $\geq 1$  time (detections), number of sampling
#occasions (num.nights), covariate values, and diet classification (Diet)
```

```
#Define the group variables
```

```
  G <- cbind(as.numeric(D$Diet=="Carnivore"),as.numeric(D$Diet=="Herbivore"))
```

```
#Define the covariates for occupancy
```

```
  X = D[,5:8]
```

```
#Define the group covariates
```

```
  XG = cbind(X*G[,1],X*G[,2])
```

```
#Define the covariates for detection
```

```
  dX = D[,9:11]
```

```
#Load the necessary libraries
```

```
  library(R2jags); library(reshape); library(plyr)
```

```
#Define the necessary arguments to run the jags command
```

```
#Load all the data including the detection array, number of sampling occasions, individual
#species sampled, total number of sampled species, and covariate information
```

```
  data <- list(D = D$Detections, N = ceiling(D[, "num.nights"]), Species =
    as.numeric(D$SppCode), n = nrow(D), nspp = max(as.numeric(D$SppCode)), X =
    X, XG = XG, dX = dX)
```

```
#Specify the initial values
```

```
  inits = function() {list(Z = as.numeric(data$D>0))}
```

```
#Specify the parameters to be monitored
```

```
  params = c("rho", "pbeta", "spbeta", "sigpbeta", "mbeta", "sigbeta", "sbeta", "gbeta",
    "psi.mean", "sigma.occ", "p.mean", "sigma.p", "alpha", "Z", "P")
```

```
#Specify the number of chains (nc), number of iterations (ni), burn-in period (nb), and thinning
#rate (nthin)
```

```
  nc = 3
```

```
  ni = 60000
```

```
  nb = 10000
```

```
  nthin = 50
```

#Write the model code to a text file called “AllMammals.txt”

```
cat(“

data {
  nBeta <- dim(X)
  nG <- dim(XG)
  nP <- dim(dX)
}

model {

# Define covariance parameter between detection and mean occupancy
rho ~ dunif(-1,1)
var.p <- sigma.p / (1.-pow(rho,2))

#Define prior distributions for occupancy parameters
alpha.mean <- log(psi.mean) - log(1-psi.mean)
psi.mean ~ dunif(0,1)
sigma.occ ~ dunif(0,10)
tau.occ <- pow(sigma.occ,-2)

#Define prior distributions for true positive detections
p.mean ~ dunif(0,1)
b <- log(p.mean) - log(1-p.mean)
sigma.p ~ dunif(0,10)
tau.p <- pow(sigma.p,-2)

#Define prior distributions for occupancy effects where nbeta is the number of occupancy
#covariates in the model, mbeta is the community-level hyper-parameter for each of the nbeta
#covariates, tbeta is the amount of variability in each of the community-level hyper-parameters,
#and sbeta is the species-specific covariate effects
  for (a in 1:nBeta[2]){
    mbeta[a] ~ dnorm(0,0.01)
    sigbeta[a] ~ dunif(0,10)
    tbeta[a] <- pow(sigbeta[a],-2)
  }
  for (i in 1:(nspp)) {
    sbeta[i,a] ~ dnorm(0,tbeta[a])
  }
}

#Define prior distributions for nG, the group-level hyper-parameters
```

```

    for (a in 1:nG[2]){
      gbeta[a] ~ dnorm(0,0.01)
    }

#Define prior distributions for detection effects where nP is the number of detection covariates in
#the model, pbeta is the community-level hyper-parameter for each of the nP covariates, tpbeta is
#the amount of variability in each of the community-level hyper-parameters, and spbeta is the
#species-specific covariate effects
  for (a in 1:nP[2]){
    pbeta[a] ~ dnorm(0,0.01)
    sigpbeta[a] ~ dunif(0,10)
    tpbeta[a] <- pow(sigpbeta[a],-2)
    for (i in 1:(nspp)) {
      spbeta[i,a] ~ dnorm(0,tpbeta[a])
    }
  }

#Define prior distributions for the occupancy and detection covariates for each species
  for (i in 1:(nspp)) {
    alpha[i] ~ dnorm(alpha.mean, tau.occ)
    mu.p[i] <- b + (rho*sigma.p /sigma.occ)*(alpha[i] - alpha.mean)
    P[i] ~ dnorm(mu.p[i], var.p)
  }

#Estimate the occupancy probability (latent Z matrix) for each species at each camera station
  for (j in 1:n) {
    logit(psi[j]) <- alpha[Species[j]] + inprod(mbeta,X[j,])+ inprod(sbeta[Species[j],],
    X[j,]) + inprod(gbeta,XG[j,])

#Estimate the detection probability for each species at each camera station
    logit(p[j]) <- P[Species[j]] + inprod(pbeta,dX[j,]) + inprod(spbeta[Species[j],],
    dX[j,])

    Z[j] ~ dbern(psi[j])
    zp[j] <- p[j]*Z[j]
    D[j] ~ dbin(zp[j], N[j])
  }
}

#Finish writing the text file
  ", file = "AllMammals.txt")

#Run the model and call the results "output"

```

```

output <- jags(data = data, inits = inits, parameters.to.save = params, model.file
              ="AllMammals.txt", n.chains =nc, n.iter =ni, n.burnin =nb, n.thin =nthin)

#See a summary of the parameter estimates
output.sum <- output$BUGSoutput$summary

#TO ESTIMATE GROUP-LEVEL HYPER-PARAMETERS:

#Define the occupancy covariate effects where mbeta is the community-level hyper-parameter,
#gbeta is the group-level hyper-parameter, and sbeta is the species-specific parameter
mbeta <- output$BUGSoutput$sims.list$mbeta
gbeta <- output$BUGSoutput$sims.list$gbeta
sbeta <- output$BUGSoutput$sims.list$sbeta

#Define the covariates and the groups
covs <- colnames(X)
sizes <- c("Omnivore","Carnivore","Herbivore")

#Create a data frame where the number of rows is equal to the number of covariates * the
#number of groups
group <- data.frame(expand.grid(covs,sizes), matrix(NA,length(covs)*length(sizes),4))
colnames(group) <- c("Factor","Group","Mean","SD","LCI","UCI")

#Create a loop estimating the reference group values
for (a in 1:length(covs)){
  group[a,3:6] <- c(mean(mbeta[,a]),sd(mbeta[,a]),quantile(mbeta[,a],
c(0.025,0.975)))
}

#Create a second loop estimating the other group values
for (a in 1:length(covs)){
  for (b in 1:(length(sizes)-1)){
    sims <- mbeta[,a] + gbeta[,((b-1)*4+a)]
    group[(4*(b)+a),3:6] <- c(mean(sims),sd(sims),quantile(sims,
c(0.025,0.975)))
  }
}

#Export the results as a table
write.table(x=group,file="Results/group.csv",sep=",")

#TO ESTIMATE SPECIES-SPECIFIC COVARIATE VALUES:

```

```

#Begin by defining the species
spec <- Spp[,1]

#Define the group levels where 1 = carnivore, 2 = herbivore, and 3 = omnivore
levels(Spp$Diet) <- levels(Spp$Diet)[c(1,2,3)]
gg <- as.numeric(Spp$Diet)

#Define the occupancy covariates and groups
covs <- colnames(X)
sizes <- c("Omnivore", "Carnivore", "Herbivore")

#Create a data frame where the number of rows is equal to the number of covariates * the
#number of species
species <- data.frame(expand.grid(covs,spec), matrix(NA,length(covs)*length(spec),4))
colnames(species) <- c("Covariate", "Species", "Mean", "SD", "LCI", "UCI")

#Re-define gbeta
gbeta <- cbind(gbeta, matrix(0,nrow(gbeta),length(covs)))

#Create a loop that will estimate species-specific values for each of the covariates
for (a in 1:length(covs)){
  for (b in 1:length(spec)){
    sims <- mbeta[,a] + gbeta[,((gg[b]-1)*4+a)] + sbeta[,b,a]
    species[(4*(b-1)+a),3:6] <- c(mean(sims), sd(sims),
    quantile(sims,c(0.025,0.975)))
  }
}

#Export the results as a table
write.table(x=species,file="Results/species.csv",sep=",")

#TO ESTIMATE SPECIES RICHNESS FOR EACH SITE:

#Begin by increasing memory to avoid errors
memory.limit(size=10000)

#Define the z matrix
z = output$BUGSoutput$sims.list$Z

#Sort the data frame based on species, study site, and diet category
d <- sort_df(merge(data.frame(ID = 1:nrow(D),D[,1:2]),data.frame(SppCode = spec,
Group = Spp$Diet)), "ID")[,c(1,3,4)]

```

```

#Create a new data frame
dz <- data.frame(d,t(z))

#Load reshape2 library
library(reshape2)

#Melt the data frame for easy casting
m.dz <- melt(dz,id.vars = c("SppCode","StudySite","Group") )

#Aggregate the data by summing the values in the z matrix for each camera station during each
#iteration
z.all <- acast(m.dz, StudySite ~ variable, fun.aggregate = sum)

#Use the aggregated values to create probability distributions and estimate mean, sd, and 95%
#credible interval values for camera-station specific species richness
z.all <- t(apply(z.all,1,function(x) c(mean(x),sd(x),quantile(x,c(0.025,0.975)))));
colnames(z.all) = c("Mean","SD","LCI","UCI")

#Export estimates of species richness as a table
write.table(x=z.all,file="Results/spprich.csv",sep=",")

#To estimate group richness for each site:
#Aggregate the data by summing the group-specific values in the z matrix for each camera
#station during each iteration
z.group <- acast(m.dz,StudySite + Group ~ variable, fun.aggregate = sum)

#Use the aggregated values to create probability distributions representing estimated camera-
#station specific group richness
z.group <- t(apply(z.group,1,function(x) c(mean(x),sd(x),quantile(x,c(0.025,0.975)))));
colnames(z.group) = c("Mean","SD","LCI","UCI")

#Export estimates of group richness as a table
write.table(x=z.group,file="Results/grouprich.csv",sep=",")

```