Appendix S2: Tutorial for creating aggregated priors using R and JAGS

This appendix is a tutorial for using prior aggregation to include external sources of information in multi-species occupancy models (MSOMs). Running the code included in this tutorial requires the software JAGS, which can be downloaded here.

This tutorial does not include general information on Bayesian MSOMs or the use and selection of ecologically informed priors. For an introduction to Bayesian MSOMs, see Chapter 11 of *Applied Hierarchical Modeling in Ecology: Analysis of Distribution, Abundance and Species Richness in R and BUGS* by Royle and Kéry (2015). For a guide to Bayesian model selection, see Hooten and Hobbs (2015). For a guide to Bayesian model checking, see Conn et al. (2018). For more information on developing ecologically informed priors, see Low Choy et al. (2009), Banner et al. (2020), and citations therein.

This tutorial requires the following packages:

```
library(R2jags)
library(boot)
library(abind)
library(tidyverse)
library(ggnewscale)

# Set seed for reproducibility:
set.seed(23)
```

1. Simulate the community

We will begin by simulating a community consisting of 10 species. We will assume that we surveyed this community by sampling 20 sites over a period of 4 surveys each. We will also simulate a covariate that is correlated with the occupancy rates of some species: half of the species in the community will respond negatively to the covariate, while the other half are not affected.

```
# Global variables
nspec <- 10 # number of species
nsite <- 20 # number of sites
nsurvey <- 4 # surveys per site

Ks <- rep(nsurvey, nsite) # vector of surveys at each site

# Vector of covariate responses: half of species respond negatively
resp2cov <- c(rnorm(n = 5, sd = 0.25),
              rnorm(n = 5, mean = -3, sd = 0.25))

resp2cov <- sample(resp2cov)

# Covariate values for sites
cov <- sort(rnorm(n = nsite))
```

To simulate site-level occupancy, we will first draw species-level occupancy probabilities from a beta distribution $\psi_i \sim Beta(\alpha = 2, \beta = 3)$. This distribution generates a wide range of occupancy probabilities (95% interval

0.067586 – 0.8058796), a situation in which data augmentation is known to work well. We will use a logit link function to account for covariate effects on site-level occupancy probability of each species. Finally, the true occupancy state for each species at each site will be the result of a Bernoulli trial with the site-level probability as the probability of success.

```
# Get probs from a beta distribution
sim.occ <- rbeta(n = nspec, shape1 = 2, shape2 = 3)

# Write function to simulate true occupancy state
tru.mats <- function(spec=nspec, site=nsite,
                     alpha1=resp2cov){

  #Get site-level psi to account for covariates
  alpha0 <- logit(sim.occ)

  #Create empty matrix to store occupancy probs
  logit.psi <- matrix(NA, nrow = spec, ncol = site)

  # Generate occupancy probs
  for(i in 1:spec){
    logit.psi[i,] <- alpha0[i] + alpha1[i]*cov
  }

  # Transform
  psi <- plogis(logit.psi)

  # Generate true occupancy state
  nlist<-list()
  for(a in 1:spec){
    nlist[[a]] <- rbinom(n = site, size = 1, prob = psi[a,])
  }

  #Turn abundance vectors into abundance matrix
  ns<-do.call(rbind, nlist)

  return(ns)
}

# Get true occupancy states
tru <- tru.mats()
```

Similarly to species-level occupancy probabilities, species-level detection probabilities will be drawn from a beta distribution $p_i \sim Beta(\alpha = 2, \beta = 8)$. This will generate low-to-mid detection probabilites (95% interval 0.028145 – 0.4824965), another situation in which data augmentation performs well. Environmental or survey covariates that may influence detectability can be added using the logit link function; however, for this example we will assume detectability does not vary across sites and surveys.

Simulated survey data will be the result of a Bernoulli trial with the species-level detection probability as the probability of encountering that species at a given site during a given survey.

```
# Generate mean detection probabilities from beta dist
mean.p <- rbeta(n = nspec, shape1 = 2, shape2 = 8)
mean.p <- sort(mean.p, decreasing = T)

# Generate detection histories
get.obs <- function(mat, specs){
```

```
  #Detection intercept and cov responses
  beta0<-logit(mean.p) #put it on logit scale

  #Logit link function
  logit.p <- array(NA, dim = c(nsite, nsurvey, specs))
  for(i in 1:specs){
    for(j in 1:nsite){
      for(k in 1:nsurvey){
        logit.p[j,,i] <- beta0[i] # Add covariates here
      }
    }
  }

  p <- plogis(logit.p)

  #Simulate observation data
  L<-list()

  for(b in 1:specs){
    y<-matrix(NA, ncol = nsite, nrow = nsurvey)
    for(a in 1:nsurvey){
      y[a,]<-rbinom(n = nsite, size = 1, prob = p[,,b]*mat[b,])
    }
    L[[b]]<-t(y)
  }

  #Smash it into array
  obs<-array(as.numeric(unlist(L)),
                dim=c(nsite, nsurvey, specs))

  return(obs)
}

obs.data <- get.obs(mat = tru, specs = nspec)

# Look at observed occurrence
maxobs <- apply(obs.data, c(1,3), max)
```

By calculating the column sums, we can see that one species went undetected in the simulated survey:

```
colSums(maxobs) # One species was not observed
```

```
##  [1] 4 4 1 5 2 7 2 0 1 1
```

To make the JAGS script easier to write and the figures more readable, the undetected species was moved to the last column in the observed data. Code for this procedure can be found in Appendix S3.

2. Define the informed prior

Next, we will define the informed species-level prior distribution for the undetected species. Although the priors can be defined in the main model text, writing them separately allows you to more easily to adjust the variance, relative weights, etc. of different prior combinations. Running models with different priors is recommended as a test for prior sensitivity.

Most Bayesian MSOMs use normally-distributed priors, but other distributions can be used. Code for aggregating non-normal distributions can be found in de Carvalho et al. (2015). We will define the mean of

informed species-level prior using the true value of the simulated covariate. We know the true value of the covariate is:

```
# Get true covariate value
resp2cov[10]
```

```
## [1] -2.745199
```

We will round this value to -3 as the mean of the informed prior distribution. We will also assign a variance of 0.5 (standard deviation of approximately 0.7). This value is somewhat arbitrary, but in general large standard deviations (> 2) are not recommended, as they can yield bimodal posterior distributions (Northrup and Gerber 2018).

We will use the Markov chain Monte Carlo (MCMC) sampler JAGS to analyze the model. JAGS is compatible with most operating systems and the language is similar to R. The package R2jags will allow us to call JAGS directly from R.

In order for JAGS to analyze the model, we have to write a text file to send to JAGS. Begin by writing a character object that defines the mean and variance of the informed prior distribution:

```
# Write script for priors in JAGS language
priors <- "#Info for species-level prior distribution
            inf.mean <- -3 #mean of distribution
            inf.var <- 0.5 #variance of distribution"
```

Next, define the relative weights of the community-level hyperprior and the informed species-level prior. The weight is a value between 0 and 1 that determines the relative contribution of each prior to the aggregated prior (weights of each prior must sum to 1). To assign weights, create a vector with the weight of the community-level prior as the first element and the species-level prior as the second:

```
priors <- paste(priors,
          "#Define prior weights: how much each distribution
          #contributes to the final aggregate
          #Hyperprior first, then informed
            weights <- c(0.5, 0.5) #these are equal weights")
```

Next, pool the distributions. For normal distributions, the pooled mean μ_{pooled} is:

$\mu_{pooled} = \sum(\mathbf{w}\mu) * v_{pooled}$

where μ is a vector of raw means and v_{pooled} the pooled variance. The pooled variance v_{pooled} is:

$v_{pooled} = 1/\sum \mathbf{w}$

The term $\mathbf{w}$ is defined as $\mathbf{w} = \alpha/\mathbf{v}$, where α is the vector of weights and $\mathbf{v}$ is a vector of raw variances.

Because $\mathbf{w}$ typically represents the regional occupancy of a species in MSOM notation, we will use the term 'lb' to calculate the pooled mean and variance. The terms 'a1.mean' and '1/tau.a1' are the mean and variance, respectively, of the community-level hyperprior, which we will define later.

```
priors <- paste(priors,
            "#Pool the distributions
            lb[1] <- weights[1]/(1/tau.a1)
            #1/tau.a0 is the variation of hyperprior
            lb[2] <- weights[2]/inf.var

            pooled.var <- 1/sum(lb)
            pooled.mean <- sum(lb*c(a1.mean,inf.mean))
              *pooled.var")
```

Finally, we will use the pooled mean and variance of the aggregated prior above when we define species-level priors:

```
priors <- paste(priors,
                "for(i in 1:spec){
                  #Create priors from hyperpriors/aggregated prior
                  w[i] ~ dbern(omega)
                  #w=1 means species was available for sampling

                  a0[i] ~ dnorm(a0.mean, tau.a0)
                  #a0 is the occupancy intercept

                  a1[i] ~ dnorm(ifelse(i==10,pooled.mean,a1.mean),
                                ifelse(i==10,(1/pooled.var),tau.a1))
                  #Use ifelse() here because detected species
                  #are still drawn from hyperprior

                  b0[i] ~ dnorm(b0.mean, tau.b0)
                  #b0 is detection intercept")
```

3. Write the JAGS script

Next, we write the full model script in the JAGS language:

```
# Function to create text file
write.model <- function(priors){
  mod <- paste("
    model{
    # Define hyperprior distributions: intercepts
    omega ~ dunif(0,1)

    mean.a0 ~ dunif(0,1)
    a0.mean <- log(mean.a0)-log(1-mean.a0)
    tau.a0 ~ dgamma(0.1, 0.1)

    mean.a1 ~ dunif(0,1)
    a1.mean <- log(mean.a0)-log(1-mean.a0)
    tau.a1 ~ dgamma(0.1, 0.1)

    mean.b0 ~ dunif(0,1)
    b0.mean <- log(mean.b0)-log(1-mean.b0)
    tau.b0 ~ dgamma(0.1, 0.1)

    ",priors,"

      #Estimate occupancy of species i at point j
      for (j in 1:J){
        logit(psi[j,i]) <- a0[i] + a1[i]*cov[j]
        Z[j,i] ~ dbern(psi[j,i]*w[i])

        #Estimate detection of i at point j during survey k
        for(k in 1:K[j]){
          logit(p[j,k,i]) <-  b0[i]
          obs[j,k,i] ~ dbern(p[j,k,i]*Z[j,i])
    }
```

```
    }
    }

    #Estimate total richness by adding observed and unobserved species
    n0<-sum(w[spec])
    N<-(spec-1)+n0

    }
    ")
  writeLines(mod, "samplemod.txt")
}

write.model(priors = priors)
```

4. Run model

Before running the model, we need to send some information to JAGS, including our data, the parameters we want JAGS to return, and the initial values for the Markov chains.

```
# List of data to send to model
datalist <- list(J = nsite, K = Ks, obs = obs.aug,
                 spec = nspec, cov = cov)

# Parameters to save after model is analyzed
parms <- c('N', 'a0', 'b0', 'a1', 'Z', 'a1.mean','tau.a1', 'pooled.mean',
           'pooled.var')

# Initial values for the Markov chains
init.values<-function(){
  maxobs <- apply(obs.aug, c(1,3), max)
  inits <- list(
    w = rep(1,nspec),
    a0 = rnorm(n = nspec),
    a1 = rnorm(n = nspec),
    b0 = rnorm(n = nspec),
    Z = maxobs)
}
```

Finally, run the model in JAGS. Additional code for saving and loading the model results can be found in Appendix S3.

5. Creating Figures

5.1 Check to see if aggregation worked We can check if prior aggregation worked by comparing the posterior distribution (i.e. the model result) to the aggregated prior, and the aggregated prior to its parent distributions. If prior aggregation was successful, the aggregated prior should be somewhere in between the informed species-level prior and the community-level hyperprior. The posterior distribution should resemble the aggregated prior more than the two parent distributions.

We will start by extracting the mean and standard deviation of each prior from the model. Note that model parameters are either variance or precision (tau); these need to be converted to standard deviation.

```
# Get values from aggregated prior
pooled.mean <- median(model$BUGSoutput$sims.list$pooled.mean)
pooled.sd <- median(sqrt(model$BUGSoutput$sims.list$pooled.var))
# Medians used because posterior is asymmetrical
```

```
# Create objects from informed values used in priors
inf.mean <- -3
inf.sd <- sqrt(1/0.5)

# Pull community distribution priors from model
comm.mean <- median(model$BUGSoutput$sims.list$a1.mean)
comm.sd <- median(sqrt(1/model$BUGSoutput$sims.list$tau.a1))
# These are symmetrical but using median for consistency

# Pull posteriors from model
post.mean <- mean(model$BUGSoutput$sims.list$a1[,10])
post.sd <- sd(model$BUGSoutput$sims.list$a1[,10])
```

We will compare the distributions using ggplot:

```
# Plot the distributions
ggplot()+
  stat_function(fun = dnorm, n = 1000,
                args = list(mean = pooled.mean, sd = pooled.sd),
                size = 1, aes(linetype = "Aggregated", color = "Prior"))+
  stat_function(fun = dnorm, n = 1000,
                args = list(mean = inf.mean, sd = inf.sd),
                size = 1, aes(linetype = "Informed", color = "Prior"))+
  stat_function(fun = dnorm, n = 1000,
                args = list(mean = comm.mean, sd = comm.sd),
                size = 1, aes(linetype = "Community", color = "Prior"))+
  stat_function(fun = dnorm, n = 1000,
                args = list(mean = post.mean, sd = post.sd),
                size = 1, aes(linetype = "Aggregated", color = "Posterior"))+
  xlim(c(-6, 5))+
  scale_linetype_manual(breaks = c("Aggregated", "Informed", "Community"),
                        values = c(1, 3, 5), name = "Prior")+
  scale_color_manual(breaks = c("Prior", "Posterior"),
                     values = c("black", "red"), name = "")+
  labs(y = "Density")+
  theme_bw(base_size = 16)+
  theme(panel.grid = element_blank(),
        axis.title.x = element_blank())
```

Based on this figure, prior aggregation was successful. The posterior distribution (red) is most similar to the aggregated prior (solid black line). Note that the posterior has been pulled slightly towards the center of the community-level prior (dashed black line): this is normal, and occurs as a result of modeling all species in the context of the community.

5.2 Regional richness estimates Next, we will evaluate whether the model successfully accounted for the regional occurrence of the undetected species. First we will extract the posterior distribution of the parameter N, or regional species richness, from the model. To determine whether the model accounted for the missing species, you can use a measure of centrality such as the median:

```
# Extract regional species richness N from model
Ns <- as.vector(model$BUGSoutput$sims.list$N)

# Create table of counts for each estimate
Ns %>%
```

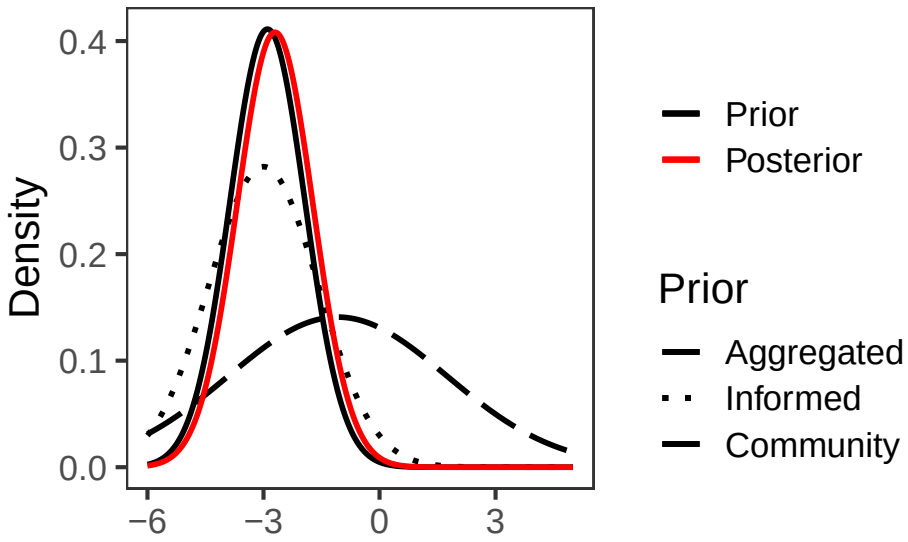

Figure 1: Comparison of the posterior distribution of the undetected species (red line) and the priors (black lines). The aggregated prior (solid black line) should fall somewhere in between the informed species-level prior (dotted black line) and uninformed community-level prior (dashed black line).

```
  table() %>%
  data.frame() %>%
  {. ->> ns.frame}
colnames(ns.frame) <- c("N_Species", "Freq")

# Look at mean and median estimates
median(Ns)
```

```
## [1] 10
```

Or, more commonly, the expected value for the parameter (i.e. the peak of the posterior probability distribution, Figure 2). For our simulated data, the expected value and median estimates agree on a regional richness estimate of 10 species.

```
# Check it graphically
Ns.median <- median(Ns)
ggplot(data = ns.frame, aes(x = as.integer(as.character(N_Species)),
                            y = Freq))+
  geom_col(width = 0.95, color = 'lightgray')+
  scale_x_discrete(limits = c(9,10))+
  labs(x = "Estimated Richness (N)", y = "Frequency")+
  scale_y_continuous(expand = c(0,0))+
  theme_classic(base_size = 14)+
  theme(axis.text.y = element_blank(),
        axis.title.y = element_blank(),
        legend.key.height = unit(40, units = 'pt'),
        aspect.ratio = 1/1)
```

5.3 Covariate responses To examine individual species' responses to the environmental covariate, we begin by extracting the parameter from the JAGS object and adding labels denoting species IDs:

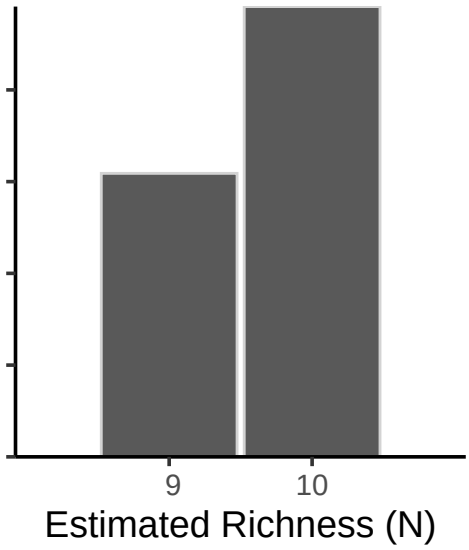

Figure 2: Posterior distribution of estimated regional species richness. The expected regional richness value, or the peak of the distribution, is 10 species, meaning the model successfully accounted for the undetected species.

```
# Extract covariate estimates from jags object
a1s <- model$BUGSoutput$sims.list$a1

a1s <- as.data.frame(a1s)

# Create a vector of species names
specnames <- logical()
for(i in 1:nspec){
  specnames[i] <- paste("Spec", i, sep = "")
}

colnames(a1s) <- specnames
```

Next, pivot the data from wide to long format for easier plotting, and calculate summary statistics. Usually, the best method for evaluating species' responses is by viewing the 95% credible interval (CI) and using the mean as the measure of centrality:

```
# Pivot data frame for plotting
a1.long <- a1s %>%
  pivot_longer(cols = everything(), names_to = "Spec",
               values_to = "a1")

a1.long$Spec <- factor(a1.long$Spec, levels = specnames)

# Get summary stats
a1.stat <- a1.long %>%
  group_by(Spec) %>%
  summarise(mean = mean(a1), lo = quantile(a1, 0.025),
            hi = quantile(a1, 0.975)) %>%
  mutate(tru.resp = resp2cov)
```

Create the plot using ggplot:

```
# Make interval plot
ggplot(data = a1.stat, aes(x = Spec, y = mean))+
  geom_point(size = 1.5)+
  geom_errorbar(ymin = a1.stat$lo, ymax = a1.stat$hi,
                size = 1, width = 0.2)+
  geom_point(aes(y = tru.resp), color = "red", size = 1.5)+
  geom_hline(yintercept = 0, linetype = "dashed", size = 1)+
  scale_y_continuous(limits = c(-25, 20))+
  labs(x = "Species", y = "Coefficient")+
  theme_bw(base_size = 14)+
  theme(panel.grid = element_blank())
```

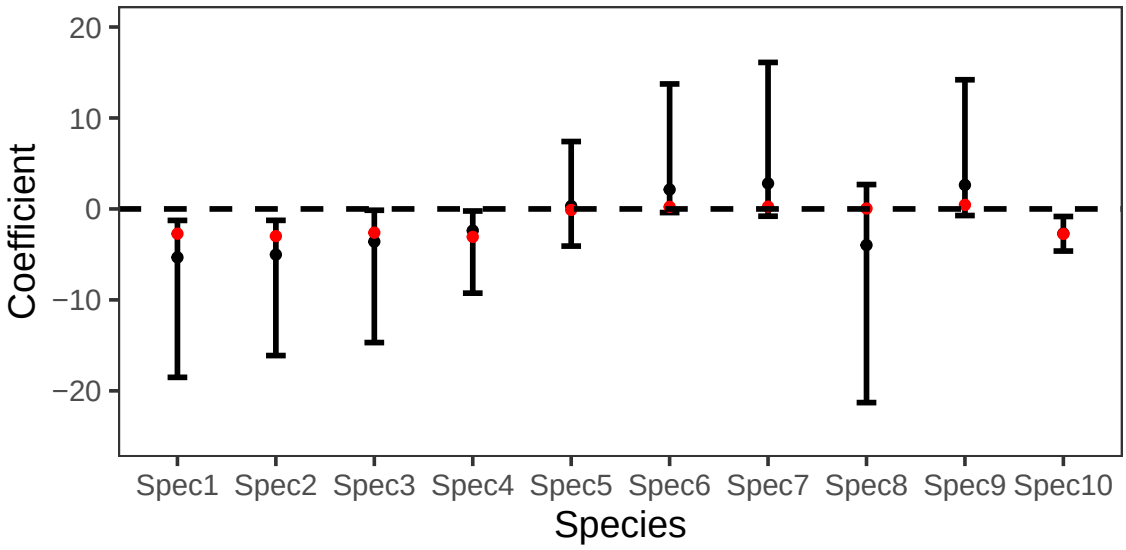

Figure 3: Estimated species-level responses to the simulated covariate. Mean estimates are denoted by black dots, whereas the true, simulated values are denoted with red dots. Error bars represent the 95% credible interval (CI); a CI which does not overlap 0 is usually considered significant.

The model correctly estimated significant negative covariate responses for detected species 1–4 and the undetected species 10 (Figure 3). Based on the position of the mean (black dots) relative to the 95% CI, we can also deduce that the posterior distributions for the detected species are highly skewed, with long tails extending away from zero. By contrast, the "stabilizing" effect of informed priors is clear in the model estimate for species 10, which has a more symmetrical and precise posterior distribution.

6. Literature Cited

Banner, K. M., K. M. Irvine, and T. J. Rodhouse. 2020. The use of Bayesian priors in Ecology: The good, the bad and the not great. Methods in Ecology and Evolution 11:882–889.

Carvalho, L. M. de, D. A. M. Villela, F. C. Coelho, and L. S. Bastos. 2015. Choosing the weights for the logarithmic pooling of probability distributions. arXiv:1502.04206 [stat].

Choy, S. L., R. O'Leary, and K. Mengersen. 2009. Elicitation by design in ecology: Using expert opinion to inform priors for Bayesian statistical models. Ecology 90:265–277.

Conn, P. B., D. S. Johnson, P. J. Williams, S. R. Melin, and M. B. Hooten. 2018. A guide to Bayesian model checking for ecologists. Ecological Monographs 88:526–542.

Hooten, M. B., and N. T. Hobbs. 2015. A guide to Bayesian model selection for ecologists. Ecological Monographs 85:3–28.

Kery, M., and J. A. Royle. 2015. Applied Hierarchical Modeling in Ecology: Analysis of distribution, abundance and species richness in R and BUGS: Volume 1:Prelude and Static Models. Academic Press.

Northrup, J. M., and B. D. Gerber. 2018. A comment on priors for Bayesian occupancy models. PLOS ONE 13:e0192819.