

Appendix S2

Matthew T. Farr, David S. Green, Kay E. Holekamp, Gary J. Roloff, and Elise F. Zipkin

Multispecies hierarchical modeling reveals variable responses of African carnivores to management alternatives

Ecological Applications

Multispecies distance sampling JAGS code

```
model{

#-----#
#-PRIORS-#
#-----#

#Gamma0
mu_s ~ dunif(0, 8)           #Mean
tau_s <- 1/(sig_s * sig_s)   #Precision
sig_s ~ dunif(0, 8)         #Variance

#Sigma
gamma1 ~ dnorm(0, 0.01)      #Effect of body size
gamma2 ~ dnorm(0, 0.01)      #Effect of region

#Alpha0
mu_a0 ~ dnorm(0, 0.01)       #Mean
tau_a0 ~ dgamma(0.1, 0.1)    #Precision
sig_a0 <- 1/sqrt(tau_a0)     #Variance

#Alpha1
mu_a1 ~ dnorm(0, 0.01)       #Mean
tau_a1 ~ dgamma(0.1, 0.1)    #Precision
sig_a1 <- 1/sqrt(tau_a1)     #Variance

#Beta1
mu_b1 ~ dnorm(0, 0.01)       #Mean
tau_b1 ~ dgamma(0.1, 0.1)    #Precision
sig_b1 <- 1/sqrt(tau_b1)     #Variance

#Overdispersion
r.N ~ dunif(0,100)           #Number of groups
r.G ~ dunif(0,100)           #Group size

for(s in social){
```

```

#Expected Group Size
beta0[s] ~ dunif(0,50)          #Intercept parameter
beta1[s] ~ dnorm(mu_b1, tau_b1) #Effect parameter

} #end s loop

for(s in 1:nspec){

#Psi
tau_p[s] ~ dgamma(0.1, 0.1) #Precision
sig_p[s] <- 1/sqrt(tau_p[s]) #Variance

#Sigma
gamma0[s] ~ dnorm(mu_s, tau_s) #Intercept parameter

#Expected Number of Groups
alpha0[s] ~ dnorm(mu_a0, tau_a0) #Intercept parameter
alpha1[s] ~ dnorm(mu_a1, tau_a1) #Effect parameter

for(j in 1:nsites){

psi[j,s] ~ dnorm(0, tau_p[s])      #Transect effect parameter

#Scale parameter
sigma[j,s] <- exp(gamma0[s] + gamma1 * size[s] + gamma2 * region[j])

#-----#
#-LIKELIHOOD-#
#-----#

for(t in 1:nreps[j]){

#Construct cell probabilities for nG cells using numerical integration
#Sum of the area (rectangles) under the detection function

for(k in 1:nD){

#Half-normal detection function at midpt (length of rectangle)
g[k,t,j,s] <- exp(-mdpt[k]*mdpt[k]/(2*sigma[j,s]*sigma[j,s]))

#Proportion of each interval (width of rectangle) for both sides of the transect
pi[k,t,j,s] <- v/B

#Detection probability for each distance class k (area of each rectangle)
f[k,t,j,s] <- g[k,t,j,s] * pi[k,t,j,s]

#Conditional detection probability (scale to 1)
fc[k,t,j,s] <- f[k,t,j,s]/pcap[t,j,s]

}#end k loop

#Detection probability at each transect (sum of rectangles)
pcap[t,j,s] <- sum(f[1:nD,t,j,s])

```

```

#Observed population @ each t,j,s (N-mixture)
y[t,j,s] ~ dbin(pcap[t,j,s], N[t,j,s])

#Latent Number of Groups @ each t,j,s (negative binomial)
N[t,j,s] ~ dpois(lambda.star[t,j,s])

#Expected Number of Groups
lambda.star[t,j,s] <- rho[t,j,s] * lambda[t,j,s]

#Overdispersion parameter for Expected Number of Groups
rho[t,j,s] ~ dgamma(r.N, r.N)

#Linear predictor for Expected Number of Groups
lambda[t,j,s] <- exp(alpha0[s] + alpha1[s] * region[j] + psi[j,s] + log(offset[j]))

}#end t loop

#Mean detection probability @ each j,s
psite[j,s] <- mean(pcap[1:nreps[j], j, s])

}#end j loop

#Mean detection probability for each species
Dprop[s] <- mean(psite[1:nsites, s])

}#end s loop

#Mean detection probability for all t,j,s
TotalDprop <- mean(Dprop[])

for(s in social){

for(j in 1:nsites){

for(t in 1:nreps[j]){

#Expected Group Size
gs.lam.star[t,j,s] <- gs.lam[t,j,s] * gs.rho[t,j,s]

#Overdispersion parameter for Expected Group Size
gs.rho[t,j,s] ~ dgamma(r.G, r.G)

#Linear predictor for Expected Group Size
gs.lam[t,j,s] <- exp(beta0[s] + beta1[s] * region[j] + log(offset[j]))

}#end t loop

}#end j loop

}#end s loop

for(i in 1:nobs){

#Observed distance classes

```

```

dclass[i] ~ dcat(fc[1:nD, rep[i], site[i], spec[i]])

}#end i loop

for(i in 1:nsoc){

#Observed Group Size (zero truncated negative binomial)
gs[i] ~ dpois(gs.lam.star[s.rep[i], s.site[i], s.spec[i]]) T(1,)

}#end i loop

for(s in social){

for(j in 1:nsites){

for(t in 1:nreps[j]){

#Abundance per transect
GSrep[t,j,s] <- lambda.star[t,j,s] * gs.lam.star[t,j,s]

}#end t loop

#Abundance per transect averaged over surveys
GSsite[j,s] <- mean(GSrep[1:nreps[j], j, s])

}#end j loop

#Mean abundance per transect
GS[s] <- mean(GSsite[1:nsites, s])

#Abundance per transect for each region
RegGS[s,1] <- mean(GSsite[1:13, s])      #Mara Triangle
RegGS[s,2] <- mean(GSsite[14:17, s])     #Talek region

}#end s loop

}#end model

```