Padilla and Sutherland R scripts

Benjamin Padilla and Chris Sutherland

Overview

This document contains the `R` scripts used to prepare data and fit multi-species N-mixture models using `nimble`.

Nimble code for multi-species N-mixture model

```
library(nimble)
library(mcmcplots)
library(MCMCvis)
library(coda)
library(abind)


model_code <- nimbleCode({

   # Community priors (with hyperparameters) for species-specific parameters ####
   for(k in 1:nspec){
      phi[k] ~ dunif(0,1)     # Zero-inflation

      # Alpha parameters (detection) ####
      for(s in 1:3){
         alpha0[k,s] ~ dnorm(mu.alpha0[s], tau.alpha0[s])        #year-specific intercept
      }
      alpha1[k] ~ dnorm(mu.alpha1, tau.alpha1)          # soft-2-hard on detection
      alpha2[k] ~ dnorm(mu.alpha2, tau.alpha2)          # green-2-broum on detection
      alpha3[k] ~ dnorm(mu.alpha3, tau.alpha3)          # Julian Day on detection
      alpha4[k] ~ dnorm(mu.alpha4, tau.alpha4)          # Time effect on detection
      alpha5[k] ~ dnorm(mu.alpha5, tau.alpha5)          # Julian Day QUAD effect on detection

      # Beta Parameters (Abundance) ####
      for(s in 1:3){
         beta0[k,s] ~ dnorm(mu.beta0[s], tau.beta0[s])        #year-specific intercept
      }
      beta1[k] ~ dnorm(mu.beta1, tau.beta1)             # soft-2-hard on abundance
      beta2[k] ~ dnorm(mu.beta2, tau.beta2)             # green-2-brown on abundance
      beta3[k] ~ dnorm(mu.beta3, tau.beta3)             # soft-2-hard Quad on abundance
      beta4[k] ~ dnorm(mu.beta4, tau.beta4)             # green-2-brown Quad on abundance
   }

   # Hyperpriors for community hyperparameters ####
```

```
# abundance model
for(s in 1:3){
   mu.beta0[s] ~ dnorm(0 , 0.01)

}
for(s in 2:3){
   sd.beta0[s] <- sd.beta0[1]
   tau.beta0[s] <- pow(sd.beta0[1], -2)
}
sd.beta0[1] ~ dunif(0 , 8)
tau.beta0[1] <- pow(sd.beta0[1], -2)

mu.beta1 ~ dnorm(0 , 0.01)
sd.beta1 ~ dunif(0 , 8)
tau.beta1 <- pow(sd.beta1, -2)
mu.beta2 ~ dnorm(0 , 0.01)
sd.beta2 ~ dunif(0 , 8)
tau.beta2 <- pow(sd.beta2, -2)
mu.beta3 ~ dnorm(0 , 0.01)
sd.beta3 ~ dunif(0 , 8)
tau.beta3 <- pow(sd.beta3, -2)
mu.beta4 ~ dnorm(0 , 0.01)
sd.beta4 ~ dunif(0 , 8)
tau.beta4 <- pow(sd.beta4, -2)

# detection model
for(s in 1:3){
   mu.alpha0[s] ~ dnorm(0 , 0.01)
   sd.alpha0[s] ~ dunif(0 , 8)
   tau.alpha0[s] <- pow(sd.alpha0[s], -2)
}
mu.alpha1 ~ dnorm(0 , 0.01)
sd.alpha1 ~ dunif(0, 8)
tau.alpha1 <- pow(sd.alpha1, -2)
mu.alpha2 ~ dnorm(0 , 0.01)
sd.alpha2 ~ dunif(0, 8)
tau.alpha2 <- pow(sd.alpha2, -2)
mu.alpha3 ~ dnorm(0 , 0.01)
sd.alpha3 ~ dunif(0, 8)
tau.alpha3 <- pow(sd.alpha3, -2)
mu.alpha4 ~ dnorm(0 , 0.01)
sd.alpha4 ~ dunif(0, 8)
tau.alpha4 <- pow(sd.alpha4, -2)
mu.alpha5 ~ dnorm(0 , 0.01)
sd.alpha5 ~ dunif(0, 8)
tau.alpha5 <- pow(sd.alpha4, -2)

# Ecological model for true abundance (process model) ####
for(k in 1:nspec){
   for (i in 1:nsite){
      a[i,k] ~ dbern(phi[k])   # zero-inflation
      N[i,k] ~ dpois(a[i,k] * lambda[i,k])
      log(lambda[i,k]) <- beta0[k, year[i]] + beta1[k] * s2h[i] + beta2[k] * g2b[i] + beta3[k] * s2h
```

```
        # Compute presence/absence matrix z (for N > 0) from latent abundance
        z[i,k] <- step(N[i,k]-1)  # returns TRUE if N >= 0
      }
   }

   # Observation model for replicated counts ####
   for(k in 1:nspec){
      for (i in 1:nsite){
         for (j in 1:nrep){
            Yc[i,j,k] ~ dbin(p[i,j,k], N[i,k])
            logit(p[i,j,k]) <- alpha0[k, year[i]] + alpha1[k] * s2h[i] + alpha2[k] * g2b[i] + alpha3[k]
         }
      }
   }

   # Other derived quantities ####
   for(k in 1:nspec){
      for(s in 1:3){
         mlambda[k,s] <- phi[k] * exp(beta0[k,s]) # Expected abundance on natural scale
         logit(mp[k,s]) <- alpha0[k,s]     # Mean detection on natural scale
      }
      Nocc.fs[k] <- sum(z[1:nsite,k])      # Number of occupied sites among the 42
   }
   for (i in 1:nsite) {
      Nrich[i] <- sum(z[i,1:nspec])        # Number of occurring species at each site
      Nbirds[i] <- sum(N[i,1:nspec])        # Number of occurring species at each site
      Nmig[i] <- sum(z[i,1:nspec] * mig[1:nspec])   # Number of migrant species
      Nres[i] <- sum(z[i,1:nspec] * res[1:nspec])   # Number of resident species
      Nsdm[i] <- sum(z[i,1:nspec] * sdm[1:nspec])   # Number of short-distance migrant species
      Ninv[i] <- sum(z[i,1:nspec] * inv[1:nspec])   # Number of insectivorous species
      Nomni[i] <- sum(z[i,1:nspec] * omni[1:nspec]) # Number of omnivorous species
      Ngran[i] <- sum(z[i,1:nspec] * gran[1:nspec]) # Number of granivorous species
      Npred[i] <- sum(z[i,1:nspec] * prd[1:nspec])  # Number of predatory species
   }
}
)
```

Code for model implementation

```
load("mod_data.Rdata")

# constants - starting values - and parameters to monitor ####
mod_constants <- list(Yc = Yc_data,                 # observed counts
                      nsite = dim(Yc_data)[1],      # 126 (42*3)
                      nrep = dim(Yc_data)[2],       # 3
                      nspec = dim(Yc_data)[3],      # 83
                      s2h = rep(sCovs$slg_1,3),     # numeric
                      g2b = rep(sCovs$slg_2,3),     # numeric
                      s2hQ = rep(sCovs$slg_1^2,3), # numeric
                      g2bQ = rep(sCovs$slg_2^2,3), # numeric
                      day = jday,           # numeric
```

```
                      dayQ = jday^2,         # numeric
                      time = time,           # numeric
                      year = year,           # numeric (1,2,3)
                      mig = mig,
                      sdm = sdm,
                      res = res,
                      inv = invs,
                      omni = omni,
                      gran = gran,
                      prd = prd)

# prepare initial values
ast <- matrix(rep(1, dim(Yc_data)[3] * dim(Yc_data)[1]), nrow = dim(Yc_data)[1]) # All species will beg
some.more <- 5
Nst <- apply(Yc_data, c(1,3), max, na.rm = T) + some.more  # initial abundance > obs abundance
inits <- function() list(a = ast,
                         phi = runif(dim(Yc_data)[3],0,1),
                         N = Nst,              # starting values for the species specific abundance
                         alpha0 = matrix(rnorm(dim(Yc_data)[3]*3),nrow = dim(Yc_data)[3],ncol = 3), # s
                         beta0 = matrix(rnorm(dim(Yc_data)[3]*3), nrow = dim(Yc_data)[3],ncol = 3), # s
                         alpha1 = rnorm(dim(Yc_data)[3]),    # soft-2-hard on detection
                         beta1 = rnorm(dim(Yc_data)[3]),    # soft-2-hard on abundance
                         alpha2 = rnorm(dim(Yc_data)[3]),    # green-2-brown on detection
                         beta2 = rnorm(dim(Yc_data)[3]),    # green-2-brown on abundance
                         alpha3 = rnorm(dim(Yc_data)[3]),    # julian day on detection
                         beta3 = rnorm(dim(Yc_data)[3]),    # soft-2-hard QUAD on abundance
                         alpha4 = rnorm(dim(Yc_data)[3]),    # time on detection
                         beta4 = rnorm(dim(Yc_data)[3]),    # green-2-brown QUAD on abundance
                         alpha5 = rnorm(dim(Yc_data)[3]),    # julian day QUAD on detection

                         # starting value for hyper-parameters means and SDs
                         mu.alpha0 = rnorm(3),
                         mu.beta0 = rnorm(3),
                         sd.beta0 = runif(3,0,3),
                         sd.alpha0 = runif(3,0,3),

                         mu.alpha1 = rnorm(1),
                         sd.alpha1 = runif(1,0,3),
                         mu.alpha2 = rnorm(1),
                         sd.alpha2 = runif(1,0,3),
                         mu.alpha3 = rnorm(1),
                         sd.alpha3 = runif(1,0,3),
                         mu.alpha4 = rnorm(1),
                         sd.alpha4 = runif(1,0,3),
                         mu.alpha5 = rnorm(1),
                         sd.alpha5 = runif(1,0,3),

                         mu.beta1 = rnorm(1),
                         sd.beta1 = runif(1,0,3),
                         mu.beta2 = rnorm(1),
                         sd.beta2 = runif(1,0,3),
                         mu.beta3 = rnorm(1),
                         sd.beta3 = runif(1,0,3),
```

```
                        mu.beta4 = rnorm(1),
                        sd.beta4 = runif(1,0,3))

parameters <- c("N","phi", "mp", "mlambda", "Nrich","Nbirds","Nmig","Nres","Nomni","Ngran","Nsdm","Ninv
                "beta0","beta1","beta2","beta3","beta4","alpha0","alpha1","alpha2","alpha3","alpha4","al
                "mu.alpha0","sd.alpha0","mu.alpha1","sd.alpha1","mu.alpha2","sd.alpha2","mu.alpha3","sd
                "mu.beta0","sd.beta0", "mu.beta1","sd.beta1","mu.beta2","sd.beta2","mu.beta3","sd.beta3

kenburns <- 25000; chains <- 3; nits <- 100000

run.time(mod_out <- nimbleMCMC(code = model_code,
                               constants = mod_constants,
                               inits = inits(),
                               monitors = parameters,
                               nburnin = kenburns,
                               nchains = chains,
                               niter = nits,
                               thin = 10,
                               samplesAsCodaMCMC = TRUE,
                               WAIC = FALSE,
                               summary = TRUE))
```