Supplementary Material.

Program R code for linear spatial capture-recapture model.

Although this program has been used by the U.S. Geological Survey (USGS), no warranty, expressed or implied, is made by the USGS or the U.S. Government as to the accuracy and functioning of the program and related program material nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

# Code for spatial linear capture-recapture model used for 2010 American shad in the Little River, NC

setwd("file path")

library(rjags)

library(reshape)

# Constants:

M <- 315 # Number of individuals

T <- 12 # Number of periods (weeks)

nantenna <- 7 # Weir, 6 antennas

antenna.loc <- c(3.7, 7.7, 13.4, 45.3, 56.4, 72.0, 77.0) # Sampling locations (rkm)

xl <- 0 # Lower boundary, river mouth

xu <- 82 # Upper boundary, Atkinson Mill Dam

# Input and format data matrix:

# y = detection history (individual (i) x antenna (j) x week(t)); first = river immigration week, last = river

# emigration week, flow1 = mean weekly gage height, flow = standardized gage height, ON = proportion of

# sampling gear functioned ### Note: R package reshape (commands melt and cast) used to organize y data

##############################################################

##### WRITE TEXT FILE WITH JAGS / BUGS SPECIFICATION #####

##############################################################

sink("filename.txt")

cat("

model { #model

# Priors

phi ~ dunif(0,1) # Survival (constant)

mu ~ dunif(-3,3) # Intercept in sigma estimate

beta ~ dunif(-3,3) # Coefficient for flow in sigma estimate

tauv ~ dunif(0,40) # Normal dist spread for correlated S's, delete for uniform model

tau <- 1/(tauv\*tauv) # Normal dist spread for correlated S's, delete for uniform model

p0 ~ dunif(0, 1) # Weekly weir detection rate

for (t in 1:T){ #t

log(sigma[t]) <- mu + beta \* flow[t] # Weekly sigma, flow covariate

sigma2[t] <- sigma[t] \* sigma[t] # Weekly sigma2

lam0[t] ~ dgamma(0.1, 0.1) # Weekly antenna baseline encounter rate

} #t

for (i in 1:M){ #m

for(t in 1:(first[i]-1)) { #t

S[i,t] <- 0 # Individual not in river, needed to follow node in JAGS

z[i,t] <- 0 # Individual not in river, needed to follow node in JAGS

} #t

# Individual’s first period

z[i,first[i]] ~ dbern(1) # Individual known to be alive at entry into study area

S[i,first[i]] ~ dunif(xl,xu) # No prior information on individual’s location

# First period, weir

D2[i,1,first[i]] <- pow(S[i,first[i]]-antenna.loc[1], 2) # Estimate activity center

w\_lam[i,1,first[i]] <- exp(-D2[i,1,first[i]]/(2\*sigma2[first[i]]))

tmp[i,1,first[i]] <- w\_lam[i,1,first[i]]\*ON[1,first[i]] # z = 1 on entry, encounters, gear functioning

y[i,1,first[i]] ~ dbern(tmp[i,1,first[i]]) # Fit Bernoulli model

# First period, antennas

for(j in 2:nantenna) { #j

D2[i,j,first[i]] <- pow(S[i,first[i]]-antenna.loc[j], 2) # Estimate activity center

lam[i,j,first[i]] <- lam0[first[i]]\*exp(-D2[i,j,first[i]]/(2\*sigma2[first[i]])) # Lam0,sigma

tmp[i,j,first[i]] <- lam[i,j,first[i]]\*ON[j,first[i]]

y[i,j,first[i]] ~ dpois(tmp[i,j,first[i]]) # Fit Poisson model

# Subsequent periods

for (t in (first[i]+1):last[i]) { #t

S[i,t] ~ dnorm(S[i,t-1], tau)T(xl,xu) # truncated normal, correlated S

# S[i,t] ~ dunif(xl,xu) # uniform, no correlation

# S[i,t] ~ dnorm(S[i,t-1], tau) # normal, correlated S

D2[i,1,t] <- pow(S[i,t]-antenna.loc[1], 2)

w\_lam[i,1,t] <- p0\*exp(-D2[i,1,t]/(2\*sigma2[t])) # weir baseline encounter rate

tmp[i,1,t] <- z[i,t]\*w\_lam[i,1,t]\*ON[1,t]

y[i,1,t] ~ dbern(tmp[i,1,t])

for(j in 2:nantenna) { #j

D2[i,j,t] <- pow(S[i,t]-antenna.loc[j], 2)

lam[i,j,t] <- lam0[t] \* exp(-D2[i,j,t]/(2\*sigma2[t]))

tmp[i,j,t] <- z[i,t]\*lam[i,j,t]\*ON[j,t]

y[i,j,t] ~ dpois(tmp[i,j,t])

} #j

phiUP[i,t] <- z[i,t-1]\*phi # Estimate overall weekly survival rate

z[i,t] ~ dbern(phiUP[i,t]) # Estimate individual alive state

} # t

# Periods after censor

for(t in (last[i]+1):T) { #t

S[i,t] <- 0 # Fish no longer in river, needed to follow node in JAGS

z[i,t] <- 0 # Fish no longer in river, needed to follow node in JAGS

} #t

} # m

} #model

", fill = TRUE)

sink()

######################################################

##### END OF TEXT FILE ############################

######################################################

#Set up data input

data1<-list(y=y, first=first, last=last, flow=flow, M=M, ON=ON, T=T, nantenna=nantenna, antenna.loc=antenna.loc)

z=matrix(NA, M, T) # Aids in initialization

for(i in 1:M){ # M

for(t in first[i]:last[i]){ # T

z[i,t] <-1

} # T

} # M

#Set initial values

inits = function() {list(z=z, phi=runif(1,0,1), p0=runif(1,0,1), lam0=runif(12,1,5), tauv=runif(1,0,30), mu=runif(1,1,3), beta=runif(1,0,1)) }

# Parameters to follow

parameters <- c("sigma", "phi", "p0", "lam0", "mu", "beta", "tauv", "S", "z")

# NOTE: following S and z requires large amount of RAM

out1 <- jags.model("filename.txt", data1, inits, n.chains=3, n.adapt=1000)

out2 <- coda.samples(out1,parameters,n.iter=20000)

# sink("filename\_results.txt") # One option to save output

# summary(out2)

# sink()