| **title** | **author** | **date** | **output** |
| --- | --- | --- | --- |
| Rerun of the analysis from Jiménez et al. 2017 on feral pig population | rerun by Roman Luštrik, code from Jiménez et al. 2017 | 20 december 2017 | html\_document |

knitr::opts\_chunk$set(echo = TRUE)

library(nimble)

library(coda)

The data has been obtained from the supplemental material from [researchgate page](https://www.researchgate.net/publication/321804457_Spatial_mark-resight_models_to_estimate_feral_pig_population_density) (needs registration) as linked by José on SECR mailing list.

This script runs only the mark-resight part, and not the SCR-transience code. Transcience code can be found on the ResearchGate supporting information tab (see link above). The run took 17 hours on my 2010 2.67 GHz CPU (Intel M620, 4 cores).

For more information on the methods, see the paper:

Jiménez, J., Higuero, R., Charre-Medellin, J. F., & Acevedo, P. (2017). Spatial mark-resight models to estimate feral pig population density. Hystrix, (September), 1–6. <http://doi.org/10.4404/hystrix-28.2-12141>

load("./data/jimenez\_et\_al\_2017\_feral\_pigs.Rdata")

fp <- FeralPigs # rename object, less typing

rm(FeralPigs)

This is the model code.

code <- nimbleCode({

# Priors

psi ~ dunif(0,1)

lam0 ~ dunif(0,5)

sigma ~ dunif(0,5)

# Marked individuals

for(i in 1:max) {

srec[i,1] ~ dunif(xlim[1], xlim[2])

srec[i,2] ~ dunif(ylim[1], ylim[2])

distmid[i,1:J] <- (srec[i,1] - X[1:J,1])^2 + (srec[i,2] - X[1:J,2])^2

lambdamid[i,1:J] <- lam0\*exp(-distmid[i,1:J]/(2\*sigma^2))

for(j in 1:J) {

yrec[i,j] ~ dpois(lambdamid[i,j]\*K)

}

}

# Unmarked part

for(i in 1:M) {

z[i] ~ dbern(psi)

s[i,1] ~ dunif(xlim[1], xlim[2])

s[i,2] ~ dunif(ylim[1], ylim[2])

dist[i,1:J] <- (s[i,1] - X[1:J,1])^2 + (s[i,2] - X[1:J,2])^2

lam[i,1:J] <- lam0\*exp(-dist[i,1:J]/(2\*sigma^2))\*z[i]

}

for(j in 1:J){

bigLambda[j] <- sum(lam[1:M,j])

for(k in 1:K){

n[j,k] ~ dpois(bigLambda[j])

}

}

Nu <- sum(z[1:M])

N <- Nu+max

D <- N/area

})

Run the model.

fp.mdl <- nimbleModel(code = code, constants = fp$constants,

data = fp$data, inits = fp$inits, check = FALSE)

fp.mdl.comp <- compileNimble(fp.mdl)

params <- c("N", "Nu", "D", "sigma", "lam0", "psi")

mcmcSCR <- configureMCMC(fp.mdl, monitors = params)

scr.mcmc <- buildMCMC(mcmcSCR)

scr.mcmc.comp <- compileNimble(scr.mcmc, project = fp.mdl)

system.time(out <- runMCMC(scr.mcmc.comp,

niter = 15000,

nburnin = 5000,

nchains = 3,

samplesAsCodaMCMC = TRUE))

Print and plot results.

save(out, file = "./data/jimenez\_et\_al\_2017\_mcmcdata.RData") # save result for later analysis, plotting

summary(mcmc.list(out))