

# LOCAL EVALUATION OF THE INDIVIDUAL STATE-SPACE (LESS)- WOLVERINES & JAGS MODEL FITTING SCRIPT

This script performs the wolverine data analysis with a SCR-LESS as presented in Milleret al. (2018). A local evaluation of the individual state-space to scale up Bayesian spatial capture recapture. Ecology and Evolution

## I.LOAD LIBRARIES AND SET WORKING DIRECTORY

```
rm(list=ls())  
library(rjags)  
library(coda)
```

Rename the appendix S2 "SourceFunction.txt" to "SourceFunction.R" Set working directory where the SourceFunctions.R is located and source the file

```
setwd("YourWorkingdirectory")  
source("WolverineData.RData")
```

## ==== 2. LOAD SCR MODELS =====

```
### ==== 2.1 SCR LESS ====  
sink("SCR-LESS.jags")  
cat("model {  
  ##-----  
  ##----- AC PLACEMENT ----##  
  ##-----##  
  for(i in 1:n.individuals){  
    sxy[i,1] ~ dunif(xy.bounds[i,1,1], xy.bounds[i,1,2])  
    sxy[i,2] ~ dunif(xy.bounds[i,2,1], xy.bounds[i,2,2])  
    pOK[i] <- habitat.mx[trunc(sxy[i,2])+1, trunc(sxy[i,1])+1]  
    OK[i] ~ dbern(pOK[i])  
  }#i  
  ##-----  
  ##----- DEMOGRAPHIC PROCESS ----##  
  ##-----##  
  psi0 ~ dunif(0,1)  
  psi <- mean(psi1[])  
  for (i in 1:n.individuals){  
    psi1[i] <- 1-(1-psi0)^prop.habitat[i]  
    z[i] ~ dbern(psi1[i])  
  }#i  
  ##-----  
  ##----- DETECTION PROCESS-----##  
  ##-----##  
  p0 ~ dunif(0,1)  
  sigma ~ dunif(0,50)
```

```

alpha <- -1/(2*sigma*sigma)
#----- DETECTION PROCESS -----#
for (i in 1:n.individuals){
  for (j in 1:n.detectors[i]){
    d2[i,j] <- pow(sxy[i,1] - detector.xy[detector.index[i,j], 1], 2) +
    pow(sxy[i,2] - detector.xy[detector.index[i,j], 2], 2)
    p[i,j] <- p0 * exp(alpha * d2[i,j])
    y[i,detector.index[i,j]] ~ dbern(p[i,j]*z[i])
  }#j
}#i
##-----
##----- DERIVED PARAMETERS -----##
##-----##
N <- sum(z[])
} ", fill = TRUE)
sink()

```

## II.SET SIMULATION PARAMETERS

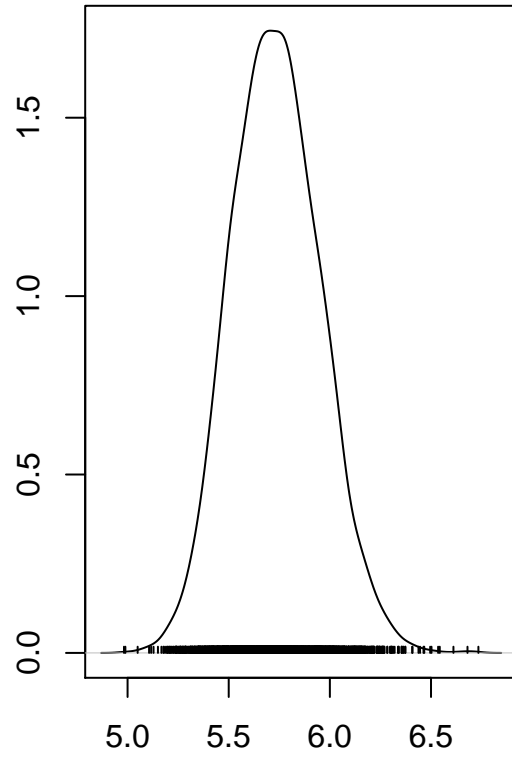
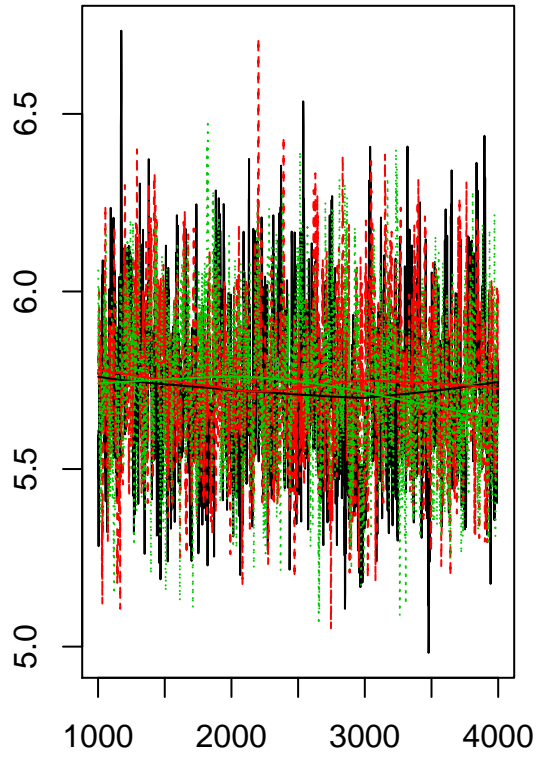
```

temp.model <- jags.model( file = "SCR-LESS.jags"
                        , data = my.jags.input
                        , inits = inits.1
                        , n.chains = 3
                        , n.adapt = 1000)
# GENERATE POSTERIOR SAMPLES
jagsoutput <- coda.samples( model = temp.model
                          , variable.names = c("p0","N","sigma")
                          , n.iter = 4000
                          , thin = 1
)

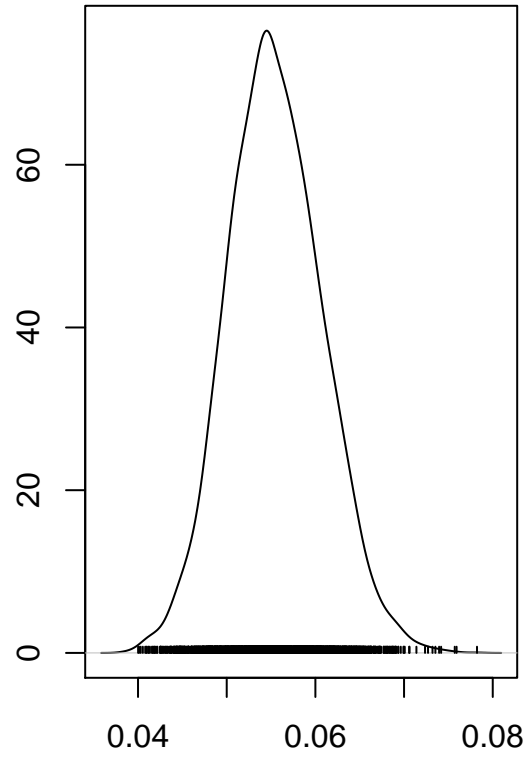
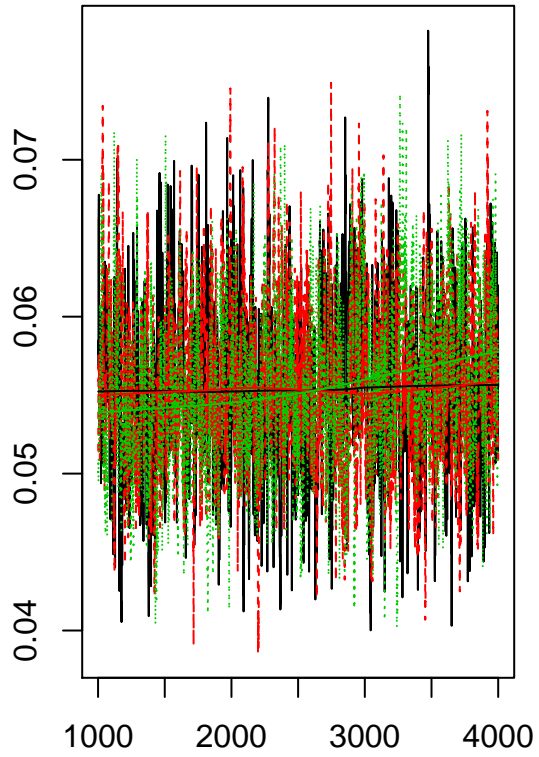
## -----
## ----- V. PLOT RESULTS JAGS MODEL -----
## -----

par(mfrow=c(1,3),mar=c(3,3,2,2))
# SIGMA
plot(jagsoutput[, "sigma"])

```



```
# p0  
plot(jagsoutput[, "p0"])
```



```
# N
plot(jagsoutput[, "N"])
```

