```
setwd("~/supplement/")  # adjust as necessary

rm(list=ls())



###
### Load R packages, subroutines, and MCMC algorithm
###

# library(lattice)

expit <- function(y){
        exp(y)/(1+exp(y))
}

source("occ.multiscale.fp.mcmc.R")  # MCMC algorithm for multi-scale false positive occupancy model



###
### Simulate multi-scale occupancy data
###

N <- 100  # number of sample units
J <- 8  # number of subunits per sample unit
K <- 5 # number of replicates per subunit

# Heterogeneity in occupancy (unit level)
X <- matrix(cbind(1,rnorm(N)),N,2)  # design matrix for occupancy
qX <- ncol(X)
```

```
beta <- matrix(c(0.5,1.5),2,1)  # coefficients for occupancy
psi <- expit(X%*%beta)  # occupancy probability
hist(psi)

# Heterogeneity in use (subunit level)
U <- cbind(1,rnorm(N*J))  # design matrix for use
qU <- ncol(U)
gamma <- matrix(c(-0.0,1),2,1)  # coefficients for use
theta <- expit(U%*%gamma)  # probability of use
hist(theta)

# Heterogeneity in detection
W <- cbind(1,rnorm(K*J*N))  # design matrix for detection
qW <- ncol(W)
alpha <- matrix(c(0.5,1),2,1)  # coefficients for detection
p <- expit(W%*%alpha)  # detection probability
hist(p)

# Assignment of rows in X, U, and W to units, subunits, and replicates
groups <- list(X=data.frame(unit=1:N),U=data.frame(unit=rep(1:N,each=J),subunit=rep(1:J,N)),
        W=data.frame(unit=rep(1:N,each=J*K),subunit=rep(rep(1:J,each=K),N),replicate=rep(1:K,J*N)))

# Create indicator variable that maps latent 'occupancy' state (z) to 'use' state (a)
z.map <- match(groups$U$unit,groups$X$unit)

# Create indicator variable that maps latent 'use' state (a) to observations (y)
a.map <- match(paste(groups$W$unit,groups$W$subunit),paste(groups$U$unit,groups$U$subunit))

# Simulate occupancy state, use state, and observations
```

```
z <- rbinom(N,1,psi)  # latent occupancy state
a <- rbinom(N*J,1,z[z.map]*theta)  # use state
y <- rbinom(N*J*K,1,a[a.map]*p)  # observations

# Examine simulated data
table(tapply(a,z.map,sum)[z==1])  # number of "used" subunits across occupied units
table(tapply(a,z.map,sum)[z==0])  # number of "used" subunits across unoccupied units

table(tapply(y,a.map,sum)[a==1])  # number of detections across "used" subunits
table(tapply(y,a.map,sum)[a==0])  # number of detections across "unused" subunits

# Add false positives to dataset
phi <- 0.09  # probability of false positive
v <- rbinom(length(y),1,phi)  # false positive indicator variables
y.tilde <- y+v  # add false positives to data set
y.tilde[y.tilde==2] <- 1

# Create ancillary negative control data set
ctrl <- list(v=rbinom(1,50,phi),M=50)


###
### Fit false positive multiscale occupancy model to data with false positives
###

start <- list(z=z,a=a,beta=beta,gamma=gamma,alpha=alpha,phi=phi)  # starting values
priors <- list(mu.beta=rep(0,qX),mu.gamma=rep(0,qU),  # prior distribution parameters
          mu.alpha=rep(0,qW),sigma.beta=2,sigma.gamma=2,sigma.alpha=2,a=1,b=1)
tune <- list(beta=0.7,gamma=0.35,alpha=0.2)  # tuning parameters
```

```
out1 <- occ.multiscale.fp.mcmc(y.tilde,ctrl,groups,W,U,X,priors,start,tune,10000,adapt=TRUE)  # fit model



###

### Examine model output

###


matplot(out1$beta,type="l")  # posterior for beta

abline(h=beta,col=1:2,lty=2)  # add truth

apply(out1$beta,2,mean)  # posterior means for beta


matplot(out1$gamma,type="l")  # posterior for gamma

abline(h=gamma,col=1:2,lty=2)  # add truth

apply(out1$gamma,2,mean)  # posterior means for gamma


matplot(out1$alpha,type="l")  # posterior for alpha

abline(h=alpha,col=1:2,lty=2)  # add truth

apply(out1$alpha,2,mean)  # posterior means for alpha


boxplot(out1$z.mean~z)  # true occupancy versus estimated occupancy


boxplot(out1$a.mean~a)  # true occupancy versus estimated occupancy


hist(out1$phi)  # posterior for phi

abline(v=phi,lty=2)  # add truth

abline(v=ctrl$v/ctrl$M,lty=2,col=2)  # add observed
```