Integrated multispecies occupancy model for bottlenose dolphins and fisheries in the Gulf of Lion, French Mediterranean Sea

Valentin Lauret, Olivier Gimenez

20/04/2021

Contents

An ounce of theory

We consider a two-species static occupancy model à la Rota et al. (2016).

Ignoring the site index, we use the following notation for the occupancy probabilities:

- ψ_{11} is the prob. that species A and species B are both present;

- ψ_{10} is the prob. that species A is present and species B is absent;
- ψ_{01} is the prob. that species A is absent and species B is present;
- ψ_{00} is the prob. that species A and species B are both absent, with avec $\psi_{11} + \psi_{10} + \psi_{01} + \psi_{00} = 1$.

The marginal probabilities of occupancy are:

- $\Pr(z_A = 1) = \Pr(\text{species A is present}) = \psi_{10} + \psi_{11}$
- $\Pr(z_B = 1) = \Pr(\text{species B is present}) = \psi_{01} + \psi_{11}$
- $\Pr(z_A = 0) = \Pr(\text{species A is absent}) = \psi_{01} + \psi_{00}$
- $\Pr(z_B = 0) = \Pr(\text{species B is absent}) = \psi_{10} + \psi_{00}$

And the conditional probabilities (reminder: $\Pr(\text{A}|\text{B}) = \Pr(\text{A and B}) / \Pr(\text{B})$):

- $\Pr(z_A = 1 | z_B = 0) = \psi_{10}/(\psi_{10} + \psi_{00}) = \Pr(\text{species A is present given species B is absent})$;
- $\Pr(z_A = 1 | z_B = 1) = \psi_{11}/(\psi_{11} + \psi_{01}) = \Pr(\text{species A is present given species B is present})$;
- $\Pr(z_B = 1 | z_A = 0) = \psi_{01}/(\psi_{01} + \psi_{00}) = \Pr(\text{species B is present given species A is absent})$;
- $\Pr(z_B = 1 | z_A = 1) = \psi_{11}/(\psi_{11} + \psi_{10}) = \Pr(\text{species B is present given species A is present})$.

In this document, we adapted the multi-species occupancy to integrate two datasets. We considered dataset `S` (e.g SAMM aerial line transects), and dataset `G` (e.g. GDEGeM boat search-encounter program). Both monitoring collected detection / non-detection about species A and B. Then, each species has a different detection probability depending on the monitoring program considered. For example, p_A^G is the probability of detecting species A by monitoring program 'g'. Then, 16 observation 'event' can occur. We coded them:

- `1` for none species detected neither by G nor S
- `2` for species A detected by G, nothing by S
- `3` for species B detected by G, nothing by S
- `4` for both species detected by G, nothing by S
- `5` for none species detected neither by G, species A detected by S
- `6` for species A detected by G, species A detected by S
- `7` for species B detected by G, species A detected by S
- `8` for both species detected by G, species A detected by S
- `9` for none species detected neither by G, species B detected by S
- `10` for species A detected by G, species B detected by S
- `11` for species B detected by G, species B detected by S
- `12` for both species detected by G, species B detected by S
- `13` for none species detected neither by G, both species detected by S
- `14` for species A detected by G, both species detected by S
- `15` for species B detected by G, both species detected by S
- `16` for both species detected by G, both species detected by S

From the 4 ecological states and the 16 observation events, we obtaine the observation process with the following 4x16 matrix.

$$t(\theta) = \begin{bmatrix}
1 & (1-p_A^G)(1-p_A^S) & (1-p_B^G)(1-p_B^S) & (1-p_B^G)(1-p_B^S)(1-p_A^S)(1-p_A^G) \\
0 & p_A^G(1-p_A^S) & 0 & (1-p_B^S)(1-p_A^S)p_A^G(1-p_B^G) \\
0 & 0 & p_B^G(1-p_B^S) & (1-p_B^S)(1-p_A^S)p_B^G(1-p_A^G) \\
0 & 0 & 0 & (1-p_B^S)(1-p_A^S)p_A^G p_B^G \\
0 & p_A^S(1-p_A^G) & 0 & p_A^S(1-p_B^S)(1-p_A^G)(1-p_B^G) \\
0 & p_A^G p_A^S & 0 & p_A^S(1-p_B^S)p_A^G(1-p_B^G) \\
0 & 0 & 0 & p_A^S(1-p_B^S)p_B^G(1-p_A^G) \\
0 & 0 & 0 & p_A^S(1-p_B^S)p_A^G p_B^G \\
0 & 0 & 0 & p_B^S(1-p_A^S)(1-p_A^G)(1-p_B^G) \\
0 & 0 & p_B^G p_B^S & p_B^S(1-p_A^S)p_A^G(1-p_B^G) \\
0 & 0 & 0 & p_B^S(1-p_A^S)p_B^G(1-p_A^G) \\
0 & 0 & 0 & p_B^S(1-p_A^S)p_B^G p_A^G \\
0 & 0 & 0 & p_A^S p_B^S(1-p_B^G)(1-p_A^G) \\
0 & 0 & 0 & p_A^S p_B^S p_A^G(1-p_B^G) \\
0 & 0 & 0 & p_A^S p_B^S p_B^G(1-p_A^G) \\
0 & 0 & 0 & p_B^G p_B^S p_A^G p_A^S
\end{bmatrix}$$

Let's go for the analysis.

Format and visualise data

Load grid and occupancy data.

SAMM

Visualise data.

Build datasets.

GDEGeM

Visualise data.

Build datasets.

Joining the dataset and formatting for Bayesian analyses

Format data.

Let's format the data in a matrix with N rows (sites) and J columns (surveys) with in each cell a `1:16` for the observation (or event in the capture-recapture terminology) `none species detected`, `species A detected`, `species B detected`, `both species detected` and each of these event by either by SAMM only, by GDEGeM onyl, or by both monitoring programs.

Do logistic regressions on detections

```
# table
table(y)

# test with logistic reg
yG[[1]][is.na(yG[[1]])] <- 0
yG[[2]][is.na(yG[[2]])] <- 0
y_turG <- apply(yG[[1]][!ind_effort,],1,max)
y_fisG <- apply(yG[[2]][!ind_effort,],1,max)

fit <- glm( y_turG ~ as_factor(y_fisG), offset = rowSums(seffG), family =binomial)
summary(fit)
```

```
# samm
yS[[1]][is.na(yS[[1]])] <- 0
yS[[2]][is.na(yS[[2]])] <- 0
y_turS <- apply(yS[[1]][!ind_effort,],1,max)
y_fisS <- apply(yS[[2]][!ind_effort,],1,max)

sum(y_fisS)
sum(y_turS)
fitS <- glm( y_turS ~ as_factor(y_fisS), offset = rowSums(seffS), family =binomial)
summary(fitS)
```

```
# both
y_tur <- y_turG + y_turS
y_fis <- y_fisG + y_fisS
y_tur[y_tur > 1] <- 1
y_fis[y_fis > 1] <- 1

sum(y_tur)
sum(y_fis)

fit <- glm( y_tur ~ as_factor(y_fis), offset = rowSums(seffS+ seffG), family =binomial)
summary(fit)
```

Full IMSO w/ constant natural parameters, and GAM sur lat/long; detection is function of sampling effort plus dolphin detection function of pres/abs of fishing boats

Get the ingredients for GAMs using package `jagam` developed by Simon Wood and basically hacks what is built by the package `mgcv`.

```
yy_dolphin <- apply(y, 1, max, na.rm = TRUE)
yy_dolphin[yy_dolphin ==1] <- 0
yy_dolphin[yy_dolphin >1] <- 1

coordx <- coordMask[,1]
coordy <- coordMask[,2]
stbathy <- mt$bathy.sc[!ind_effort,1]
library(mgcv)
res <- jagam(yy_dolphin ~ s(stbathy) + s(coordx, coordy, bs = "gp"),
             family = "binomial",
             file = "psi.txt") # same structure for fishing boats and both together
#save(res, file = 'jagam.RData')
```

We do it in NIMBLE because the MCMC run is 10 times faster than that of JAGS.

BUGS model

```
IMSOfull <- nimbleCode({

  ## state process
  for(j in 1:nsite){
    z[j] ~ dcat(psi[j, 1:4])
  }
  # occupancy probabilities

    psi[1:nsite, 1] <- 1 / (1 + sum(prop[1:nsite, 1:3])) # unoccupied
    psi[1:nsite, 2] <- prop[1:nsite, 1] / (1 + sum(prop[1:nsite, 1:3])) # occupied by species A and not
    psi[1:nsite, 3] <- prop[1:nsite, 2] / (1 + sum(prop[1:nsite, 1:3])) # occupied by species B and not
    psi[1:nsite, 4] <- prop[1:nsite, 3] / (1 + sum(prop[1:nsite, 1:3])) # occupied by both species A an

  ## observation process
  for(j in 1:nsite) {
    for(k in 1:nyear) {
      y[j, k] ~ dcat(obs[j, k, 1:16, z[j]])
    }
  }

  # detection matrix with obs for observations and state = true states
  # obs take values:
  ## VL: INSANE
  ## Let's have a bit of explanation, we have species A, and B. We have monitoring S through SAMM, and
  ## Then, we have 16 different situations
  # 1 for none species detected neither by G nor S
  # 2 for species A detected by G, nothing by S
  # 3 for species B detected by G, nothing by S
  # 4 for both species detected by G, nothing by S

  # 5 for none species detected neither by G, species A detected by S
  # 6 for species A detected by G, species A detected by S
  # 7 for species B detected by G, species A detected by S
  # 8 for both species detected by G, species A detected by S

  # 9 for none species detected neither by G, species B detected by S
  # 10 for species A detected by G, species B detected by S
  # 11 for species B detected by G, species B detected by S
  # 12 for both species detected by G, species B detected by S

  # 13 for none species detected neither by G, both species detected by S
  # 14 for species A detected by G, both species detected by S
  # 15 for species B detected by G, both species detected by S
  # 16 for both species detected by G, both species detected by S

  # given state = unoccupied,
  for(j in 1:nsite) {
    for(k in 1:nyear) {
      # state 1 = no species use the site
      obs[j, k, 1,  1] <- 1 # prob obs = 1
      obs[j, k, 2,  1] <- 0 # prob obs = 2
      obs[j, k, 3,  1] <- 0 # prob obs = 3
      obs[j, k, 4,  1] <- 0 # prob obs = 4
      obs[j, k, 5,  1] <- 0 # prob obs = 5
      obs[j, k, 6,  1] <- 0 # prob obs = 6
      obs[j, k, 7,  1] <- 0 # prob obs = 7
      obs[j, k, 8,  1] <- 0 # prob obs = 8
      obs[j, k, 9,  1] <- 0 # prob obs = 9
      obs[j, k, 10, 1] <- 0 # prob obs = 10
      obs[j, k, 11, 1] <- 0 # prob obs = 11
      obs[j, k, 12, 1] <- 0 # prob obs = 12
      obs[j, k, 13, 1] <- 0 # prob obs = 13
      obs[j, k, 14, 1] <- 0 # prob obs = 14
      obs[j, k, 15, 1] <- 0 # prob obs = 15
      obs[j, k, 16, 1] <- 0 # prob obs = 16

      # given state 2 = occupied by species A and not B,
      obs[j, k, 1,  2] <- 1 - pAg[j,k] - pAs[j,k] + pAg[j,k] * pAs[j,k] # prob obs = 1
      obs[j, k, 2,  2] <- pAg[j,k] * (1 - pAs[j,k]) # prob obs = 2
      obs[j, k, 3,  2] <- 0 # prob obs = 3
      obs[j, k, 4,  2] <- 0 # prob obs = 4
      obs[j, k, 5,  2] <- pAs[j,k] * (1 - pAg[j,k]) # prob obs = 5
      obs[j, k, 6,  2] <- pAs[j,k] * pAg[j,k] # prob obs = 6
      obs[j, k, 7,  2] <- 0 # prob obs = 7
      obs[j, k, 8,  2] <- 0 # prob obs = 8
      obs[j, k, 9,  2] <- 0 # prob obs = 9
      obs[j, k, 10, 2] <- 0 # prob obs = 10
      obs[j, k, 11, 2] <- 0 # prob obs = 11
      obs[j, k, 12, 2] <- 0 # prob obs = 12
      obs[j, k, 13, 2] <- 0 # prob obs = 13
```

```
     obs[j, k, 14, 2] <- 0 # prob obs = 14
     obs[j, k, 15, 2] <- 0 # prob obs = 15
     obs[j, k, 16, 2] <- 0 # prob obs = 16

     # given state 3 = occupied by species B and not A,
     obs[j, k, 1,  3] <- 1 - pBg[j,k] - pBs[j,k] + pBg[j,k] * pBs[j,k] # prob obs = 1
     obs[j, k, 2,  3] <- 0 # prob obs = 2
     obs[j, k, 3,  3] <- pBg[j,k] * (1 - pBs[j,k]) # prob obs = 3
     obs[j, k, 4,  3] <- 0 # prob obs = 4
     obs[j, k, 5,  3] <- 0 # prob obs = 5
     obs[j, k, 6,  3] <- 0 # prob obs = 6
     obs[j, k, 7,  3] <- 0 # prob obs = 7
     obs[j, k, 8,  3] <- 0 # prob obs = 8
     obs[j, k, 9,  3] <- pBs[j,k] * (1 - pBg[j,k]) # prob obs = 9
     obs[j, k, 10, 3] <- 0 # prob obs = 10
     obs[j, k, 11, 3] <- pBs[j,k] * pBg[j,k] # prob obs = 11
     obs[j, k, 12, 3] <- 0 # prob obs = 12
     obs[j, k, 13, 3] <- 0 # prob obs = 13
     obs[j, k, 14, 3] <- 0 # prob obs = 14
     obs[j, k, 15, 3] <- 0 # prob obs = 15
     obs[j, k, 16, 3] <- 0 # prob obs = 16

     # given state 4 = occupied by both species B and A,
     # VL: Now it's fun...
     obs[j, k, 1,  4] <- (1 - pAs[j,k]) * (1 - pAg[j,k]) * (1- pBs[j,k]) * (1 - pBg[j,k]) # prob obs =
     obs[j, k, 2,  4] <- (1 - pAs[j,k]) * (1 - pBs[j,k]) * pAg[j,k] * (1 - pBg[j,k]) # prob obs = 2
     obs[j, k, 3,  4] <- (1 - pAs[j,k]) * (1 - pBs[j,k]) * pBg[j,k] * (1 - pAg[j,k]) # prob obs = 3
     obs[j, k, 4,  4] <- (1 - pAs[j,k]) * (1 - pBs[j,k]) * pAg[j,k] * pBg[j,k]   # prob obs = 4
     obs[j, k, 5,  4] <- pAs[j,k]*(1 - pBs[j,k]) * (1 - pAg[j,k]) * (1 - pBg[j,k]) ### # prob obs = 5
     obs[j, k, 6,  4] <- pAs[j,k]*(1 - pBs[j,k]) * pAg[j,k] * (1 - pBg[j,k])  # pr ### # prob obs = 6
     obs[j, k, 7,  4] <- pAs[j,k]*(1 - pBs[j,k]) * pBg[j,k] * (1 - pAg[j,k]) # pro ### # prob obs = 7
     obs[j, k, 8,  4] <- pAs[j,k]*(1 - pBs[j,k]) * pAg[j,k] * pBg[j,k]  # prob obs ### # prob obs = 8
     obs[j, k, 9,  4] <- pBs[j,k]*(1 - pAs[j,k]) * (1 - pAg[j,k]) * (1 - pBg[j,k])  # prob obs = 9
     obs[j, k, 10, 4] <- pBs[j,k]*(1 - pAs[j,k]) * pAg[j,k] * (1 - pBg[j,k])  # prob obs = 10
     obs[j, k, 11, 4] <- pBs[j,k]*(1 - pAs[j,k]) * pBg[j,k] * (1 - pAg[j,k]) # prob obs = 11
     obs[j, k, 12, 4] <- pBs[j,k]*(1 - pAs[j,k]) * pAg[j,k] * pBg[j,k]  # prob obs = 12
     obs[j, k, 13, 4] <- pAs[j,k] * pBs[j,k] * (1 - pAg[j,k]) * (1 - pBg[j,k]) # prob obs = 13
     obs[j, k, 14, 4] <- pAs[j,k] * pBs[j,k] * pAg[j,k] * (1 - pBg[j,k]) # prob obs = 14
     obs[j, k, 15, 4] <- pAs[j,k] * pBs[j,k] * pBg[j,k] * (1 - pAg[j,k]) # prob obs = 15
     obs[j, k, 16, 4] <- pAs[j,k] * pAg[j,k] * pBs[j,k] * pBg[j,k] # prob obs = 16
   }
 }
## priors for...
# occupancy probabilities

for(j in 1:nsite) {
   log(prop[j, 1]) <- theta1[j]
   log(prop[j, 2]) <- theta2[j]
   log(prop[j, 3]) <- theta3[j]
 }

 theta1[1:nsite] <- X[1:nsite,1:33] %*% b1[1:33] ## linear predictor
 theta2[1:nsite] <- X[1:nsite,1:33] %*% b2[1:33] ## linear predictor
 theta3[1:nsite] <- X[1:nsite,1:33] %*% b3[1:33] ## linear predictor

  b1[1] ~ dnorm(0,0.01)
  b2[1] ~ dnorm(0,0.01)
  b3[1] ~ dnorm(0,0.01)

 ## prior for s(bathy)
 K11[1:9,1:9] <- S1[1:9,1:9] * lambda[1, 1]  + S1[1:9,10:18] * lambda[2, 1]
 K12[1:9,1:9] <- S1[1:9,1:9] * lambda[1, 2]  + S1[1:9,10:18] * lambda[2, 2]
 K13[1:9,1:9] <- S1[1:9,1:9] * lambda[1, 3]  + S1[1:9,10:18] * lambda[2, 3]
 b1[2:10] ~ dmnorm(zero[2:10], K11[1:9,1:9])
 b2[2:10] ~ dmnorm(zero[2:10], K12[1:9,1:9])
 b3[2:10] ~ dmnorm(zero[2:10], K13[1:9,1:9])

 ## prior for s(coordx,coordy)
 K21[1:32,1:32] <- S2[1:32,1:32] * lambda[3, 1]  + S2[1:32,33:64] * lambda[4, 1]
 K22[1:32,1:32] <- S2[1:32,1:32] * lambda[3, 2]  + S2[1:32,33:64] * lambda[4, 2]
 K23[1:32,1:32] <- S2[1:32,1:32] * lambda[3, 3]  + S2[1:32,33:64] * lambda[4, 3]
 b1[11:42] ~ dmnorm(zero[11:42], K21[1:32,1:32])
 b2[11:42] ~ dmnorm(zero[11:42], K22[1:32,1:32])
 b3[11:42] ~ dmnorm(zero[11:42], K23[1:32,1:32])

 ## smoothing parameter priors
 for (i in 1:4) {
  for (kk in 1:3){
   lambda[i, kk] ~ dgamma(.05,.005)
   rho[i, kk] <- log(lambda[i, kk])
  }
 }
 # detection probabilities (pA function of pres/abs of B, as in Waddle et al 2010 page 1470)
 # VL: There are four detections probabilities now pAs, pAg, pBg, pBs
 for(j in 1:nsite) {
   B_present[j] <- equals(z[j],3) + equals(z[j],4) #psi[j, 3] + psi[j, 4]

  for(k in 1:nyear) {
  pAs[j, k] <- (1/(1 + exp(-(beta[1] + beta[2] * B_present[j] + beta[3] * effS[j, k]))))*effindS[j,k]

  pBs[j, k] <- (1/(1 + exp(-(beta[4] + beta[5] * effS[j, k]))))*effindS[j,k]

 pAg[j, k] <- (1/(1 + exp(-(beta[6] + beta[7] *  B_present[j] + beta[8] * effG[j, k])))) *effindG[j,k]

  pBg[j, k] <- (1 /(1+ exp(-(beta[9] + beta[10] * effG[j, k]))))*effindG[j,k]

  }
 }
 for (i in 1:10){
  beta[i] ~ dnorm(0,1)
 }
})
```

Bundle data

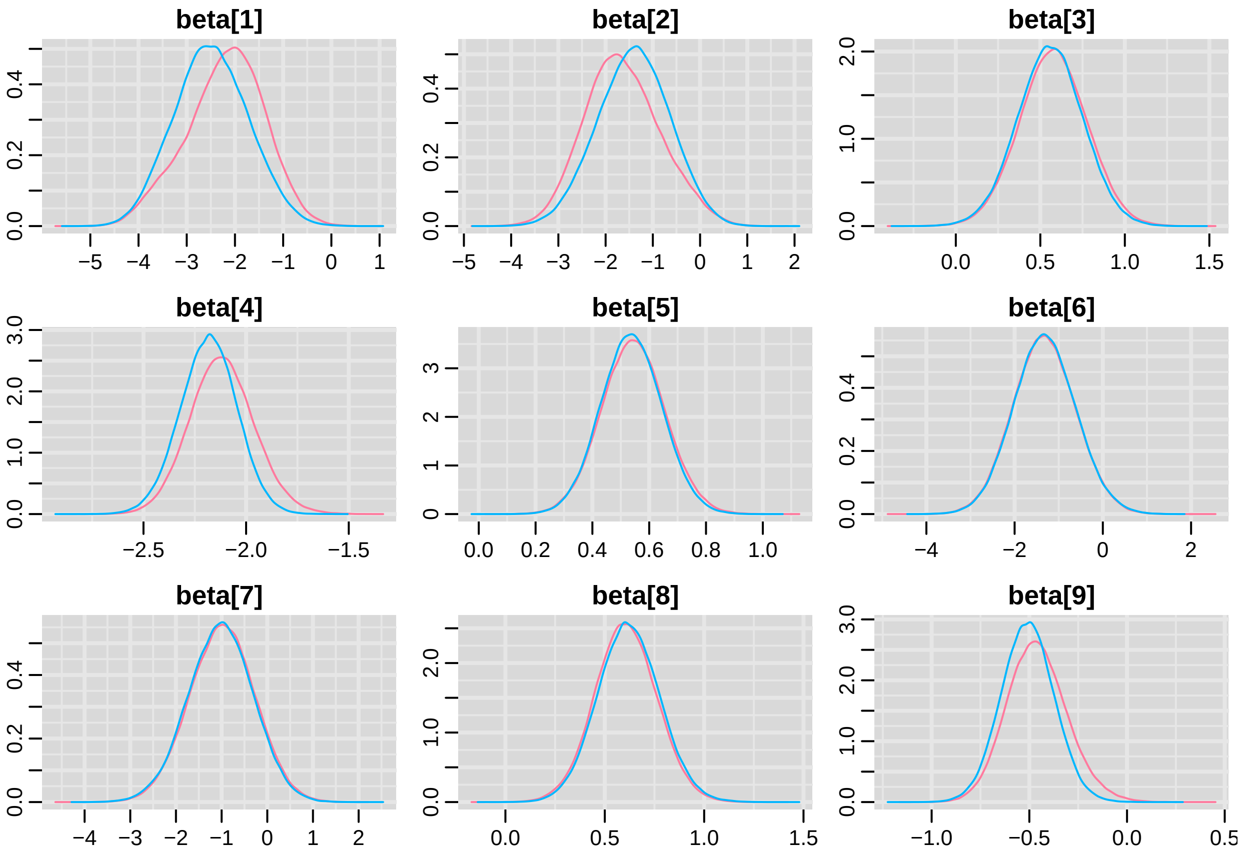
Specify data, initial values, parameters to be monitored and various MCMC details:

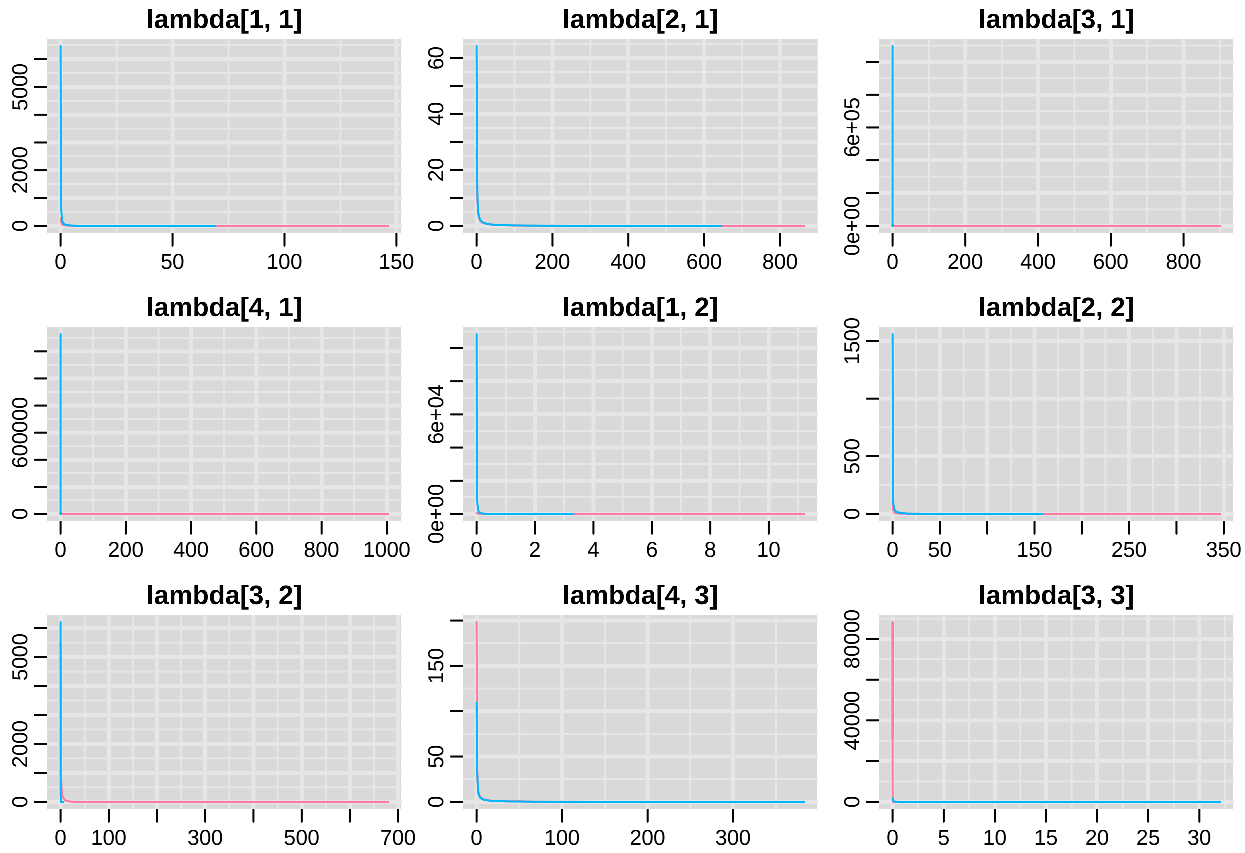
Build model

```
Rmodel <- nimbleModel(IMSOfull, constants, data, inits)
Rmodel$initializeInfo()
Rmodel$calculate() # _1736.923

conf <- configureMCMC(Rmodel)
conf$printMonitors()
conf$addMonitors("z")

conf$printSamplers(byType= TRUE)

  # RWblock sampler
  conf$removeSampler(target = c("beta"))
  conf$addSampler(target = c("beta[1]","beta[2]","beta[3]"), type = "RW_block")
  conf$addSampler(target = c("beta[4]","beta[5]"), type = "RW_block")
  conf$addSampler(target = c("beta[6]","beta[7]","beta[8]"), type = "RW_block")
  conf$addSampler(target = c("beta[9]","beta[10]"), type = "RW_block")

  conf$printSamplers(byType= TRUE)


  # Build and compile MCMC
  Rmcmc <- buildMCMC(conf)
  Cmodel <- compileNimble(Rmodel)
  Cmcmc <- compileNimble(Rmcmc, project = Cmodel)

   t <- system.time(samples2 <- runMCMC(Cmcmc, niter = 1000000, nburnin = 40000, nchains = 2, thin = 5,
   #ART nit = 5000 nb = 1000, nc = 3, t = 185 sec

str(samples2)
code = Rmodel$getCode()
save(samples2, code, file ="IMSO_1Mit.rdata")
```

Deal with output

Convergence checkings


beta[1]
0.0
0.2
0.4
−5
−4
−3
−2
−1
0
1
beta[2]
0.0
0.2
0.4
−5
−4
−3
−2
−1
0
1
2
beta[3]
0.0
1.0
2.0
0.0
0.5
1.0
1.5
beta[4]
0.0
1.0
2.0
3.0
−2.5
−2.0
−1.5
beta[5]
0
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0
beta[6]
0.0
0.2
0.4
−4
−2
0
2
beta[7]
0.0
0.2
0.4
−4
−3
−2
−1
0
1
2
beta[8]
0.0
1.0
2.0
0.0
0.5
1.0
1.5
beta[9]
0.0
1.0
2.0
3.0
−1.0
−0.5
0.0
0.5



lambda[1, 1]

lambda[2, 1]

lambda[3, 1]

lambda[4, 1]

lambda[1, 2]

lambda[2, 2]

lambda[3, 2]

lambda[4, 3]

lambda[3, 3]

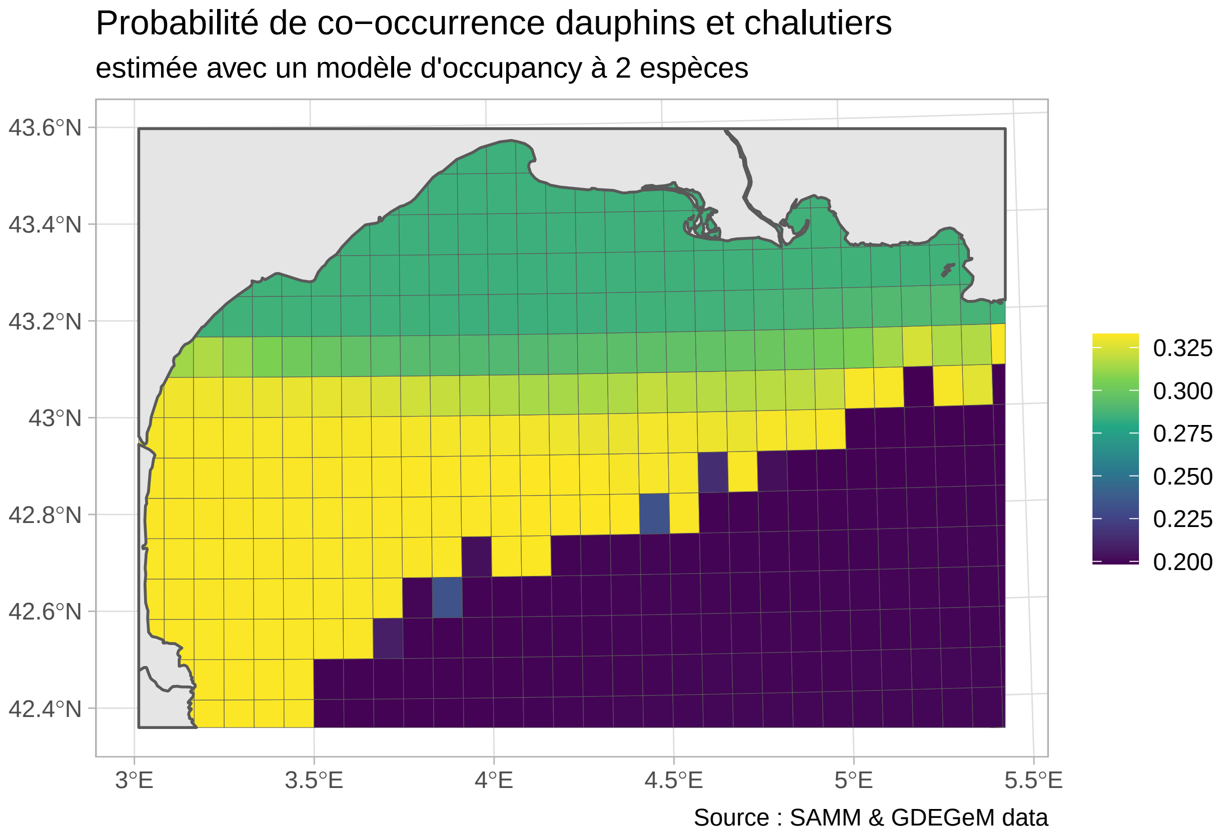
First, get the whole grid.

Second, get linear predictor.

Third, get occupancy probabilities.

Plot figures

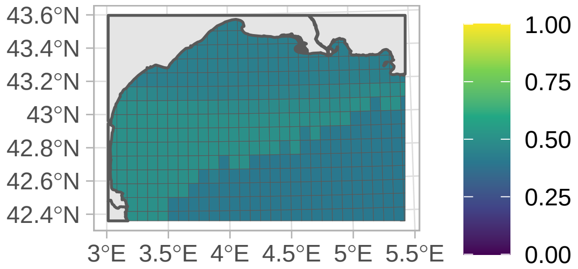
The map of coocurrences probabilities


Probabilité de co−occurrence dauphins et chalutiers
estimée avec un modèle d'occupancy à 2 espèces
43.6°N
43.4°N
43.2°N
43°N
42.8°N
42.6°N
42.4°N
3°E
3.5°E
4°E
4.5°E
5°E
5.5°E
0.325
0.300
0.275
0.250
0.225
0.200
Source : SAMM & GDEGeM data

Maps of dolphin and trawlers space-use probabilities

Bottlenose dolphins space−use prob.

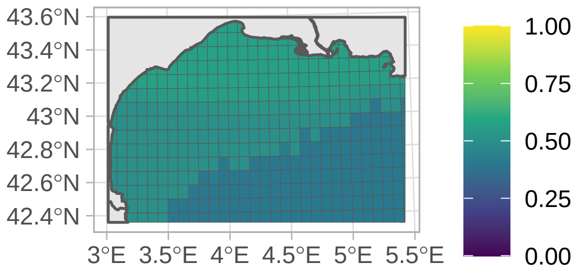
psi3 + psi1



Source : SAMM & GDEGeM data

Trawlers space−use prob.

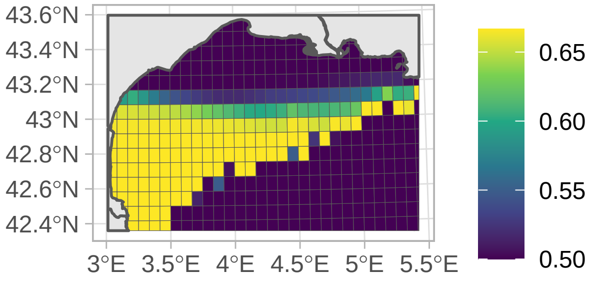
psi3 + psi2



Source : SAMM & GDEGeM data

Prob. dauphin cond. trawlers

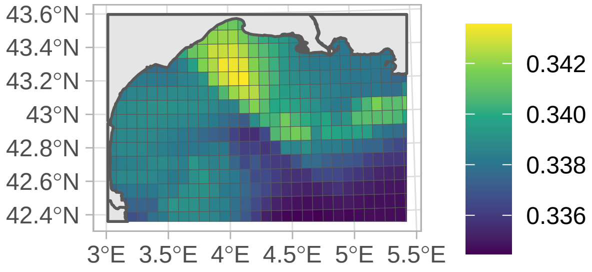
psi3 / (psi3 +psi2)



Source : SAMM & GDEGeM data

Prob. dauphin cond. no trawlers

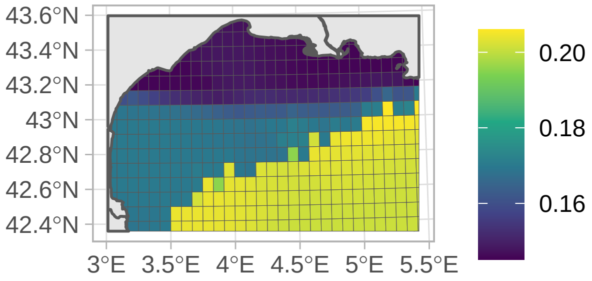
psi1 / (psi1 +psi0)



Source : SAMM & GDEGeM data

Prob. dauphin seul

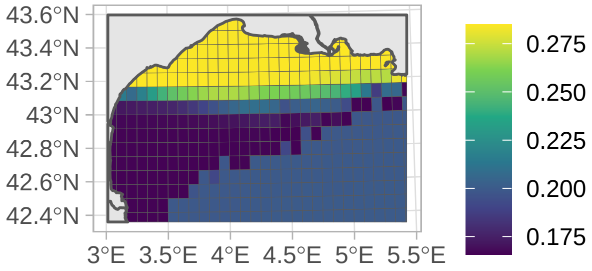
psi1



Source : SAMM & GDEGeM data

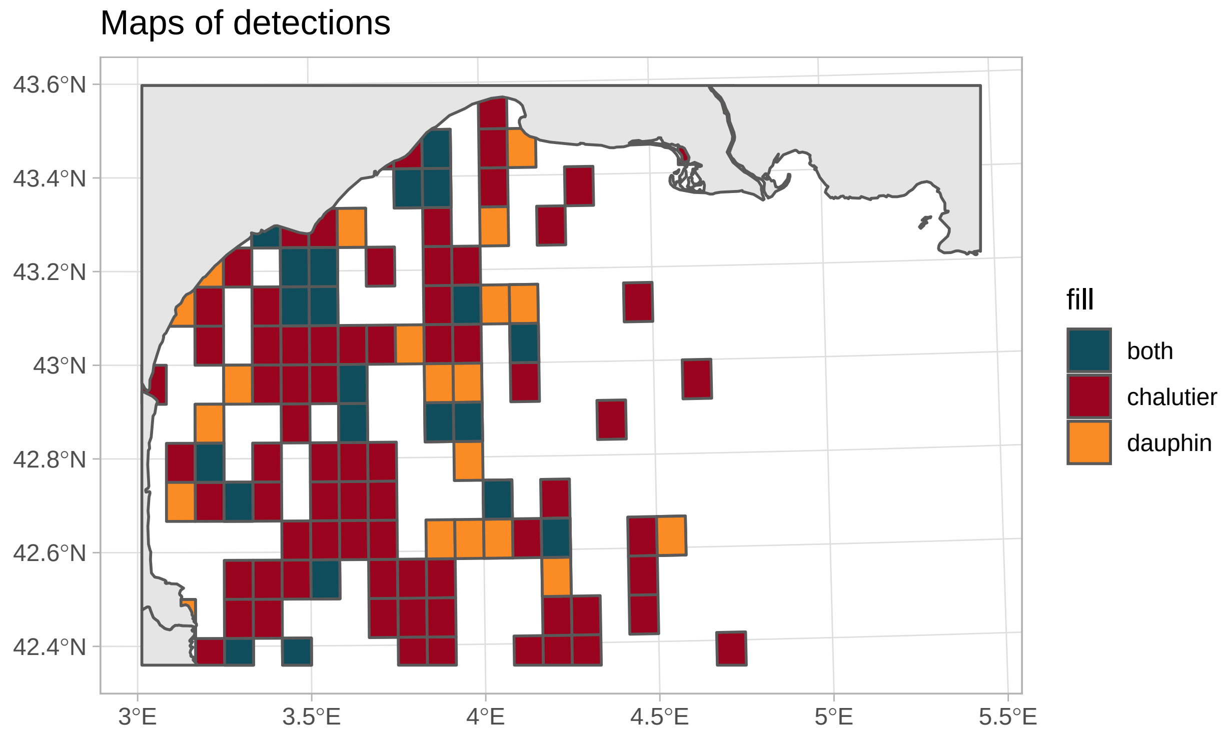
Prob. trawlers seul

psi2



Source : SAMM & GDEGeM data

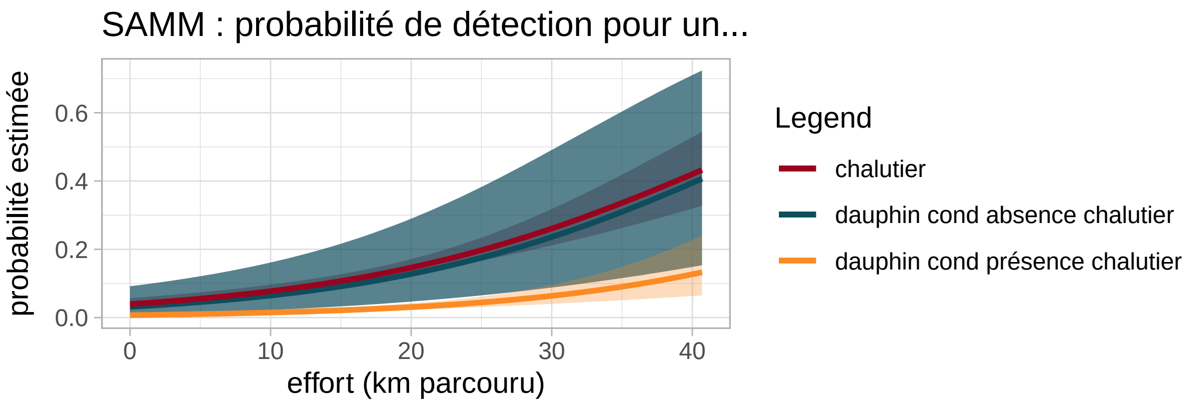
Maps of detections

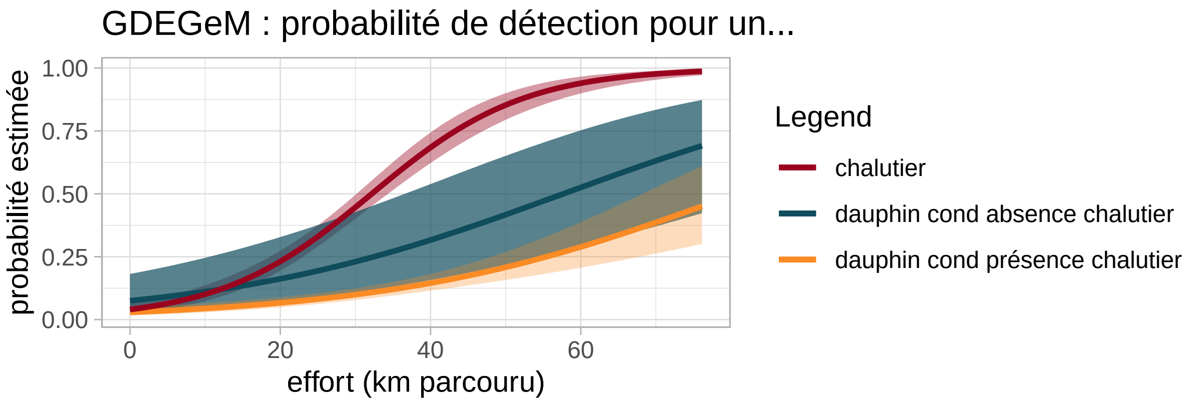


The probability of detecting dolphins conditionnal on trawlers occurence

For SAMM

For GDEGeM


SAMM : probabilité de détection pour un...
probabilité estimée
0.0
0.2
0.4
0.6
0
10
20
30
40
effort (km parcouru)
Legend
chalutier
dauphin cond absence chalutier
dauphin cond présence chalutier


GDEGeM : probabilité de détection pour un...
probabilité estimée
0.00
0.25
0.50
0.75
1.00
0
20
40
60
effort (km parcouru)
Legend
chalutier
dauphin cond absence chalutier
dauphin cond présence chalutier