Supporting Information. Tourani, M., P. Dupont, M.A. Nawaz, and R. Bischof. 2020. Multiple observation processes in spatial capture–recapture models: How much do we gain? Ecology.

Appendix S1: Model definitions in NIMBLE

Model 1 is the basic SCR using identified detections e.g., DNA samples with individual genotypes. Model 2 integrates identified and unidentified detections of the same detector type. Model 3 integrates identified detections of one detector type, with unidentified detections of the second detector type and model 4 integrates all available data including identified detections of one detector type with unidentified detections of two survey methods. An example R script is available: http://doi.org/10.5281/zenodo.3647866 to go through the model fitting process on simulated data sets.

```
Model_1 <- nimbleCode({
   ##-------------------------------------------------------------------
-----------------------------
## AC LOCATIONS
    for (i in 1:M) {
        sxy[i, 1] ~ dunif(0, x.max)                    ##equation (1)
        sxy[i, 2] ~ dunif(0, y.max)
    }

## INDIVIDUAL INCLUSION

    psi ~ dunif(0, 1)
    for (i in 1:M) {
        z[i] ~ dbern(psi)                              ##equation (2)
    }

N <- sum(z[1:M])                                       ##equation (3)


    sigma ~ dunif(0, 100)
    p0_1 ~ dunif(0, 1)
```

```
## IDENTIFIED DETECTIONS, SURVEY TYPE 1
    for (i in 1:M) {
d_squared_1[i, 1:J_1] <- (sxy[i, 1] - detector.xy_1[1:J_1,1])^2 +
(sxy[i, 2] - detector.xy_1[1:J_1,2])^2

p_1[i, 1:J_1] <- p0_1 * exp(-d_squared_1[i,1:J_1])/(2*sigma*sigma))
                                                    ##equation (4)
##p_1 in model 1 and 3 is equivalent to p_1*alpha in model 2 and 4

y_1[i, 1:J_1] ~ dbern_vector(p_1[i, 1:J_1], z[i])       ##equation (6)
    } #i

})


Model_2 <- nimbleCode({
   ##------------------------------------------------------------------
------------------------------
## AC LOCATIONS
    for (i in 1:M) {
        sxy[i, 1] ~ dunif(0, x.max)                 ## equation (1)
        sxy[i, 2] ~ dunif(0, y.max)
    }

## INDIVIDUAL INCLUSION
    psi ~ dunif(0, 1)
    for (i in 1:M) {
        z[i] ~ dbern(psi)                           ## equation (2)
    }

   N <- sum(z[1:M])                                 ## equation (3)


    sigma ~ dunif(0, 100)
    p0_1 ~ dunif(0, 1)
    alpha_1 ~ dunif(0, 1)


## SURVEY TYPE 1: IDENTIFIED + UNIDENTIFIED
    for (i in 1:M) {
d_squared_1[i, 1:J_1] <- (sxy[i, 1] - detector.xy_1[1:J_1,1])^2 +
(sxy[i, 2] - detector.xy_1[1:J_1,2])^2
p_1[i, 1:J_1] <- p0_1 * exp(-  d_squared_1[i,1:J_1])/(2*sigma*sigma))
                                                     ## equation (4)
##p_1 in model 1 and 3 is equivalent to p_1*alpha in model 2 and 4

y_1[i, 1:J_1] ~ dbern_vector(p_1[i, 1:J_1] * alpha_1, z[i]) ##
equation (6)
punid_1[i, 1:J_1] <- p_1[i, 1:J_1] * (1 - alpha_1) * z[i]
    }
    for (j in 1:J_1) {
        pdot_1[j] <- 1 - prod((1 - punid_1[1:M, j]))   ## equation (5)
```

```
    }
    ydot_1[1:J_1] ~ dbern_vector(pdot_1[1:J_1], 1)         ## equation
(7)

   })

Model_3 <- nimbleCode({
   ##-------------------------------------------------------------------
-----------------------------
   ## INDIVIDUAL INCLUSION
## AC LOCATIONS
    for (i in 1:M) {
        sxy[i, 1] ~ dunif(0, x.max)            ## equation (1)
        sxy[i, 2] ~ dunif(0, y.max)
    }


    psi ~ dunif(0, 1)
    for (i in 1:M) {
         z[i] ~ dbern(psi)                     ## equation (2)
    }

    N <- sum(z[1:M])                           ## equation (3)


    sigma ~ dunif(0, 100)
    p0_1 ~ dunif(0, 1)
    p0_2 ~ dunif(0, 1)



## IDENTIFIED DETECTIONS, SURVEY TYPE 1
    for (i in 1:M) {
d_squared_1[i, 1:J_1] <- (sxy[i, 1] - detector.xy_1[1:J_1,1])^2 +
(sxy[i, 2] - detector.xy_1[1:J_1,2])^2

p_1[i, 1:J_1] <- p0_1 * exp(-d_squared_1[i,1:J_1])/(2*sigma*sigma))
## equation (4)
##p_1 in model 1 and 3 is equivalent to p_1*alpha in model 2 and 4

y_1[i, 1:J_1] ~ dbern_vector(p_1[i, 1:J_1], z[i])    ## equation (6)
    }


## UNIDENTIFIED DETECTIONS, SURVEY TYPE 2
    for (i in 1:M) {
d_squared_2[i, 1:J_2] <- (sxy[i, 1] - detector.xy_2[1:J_2,1])^2 +
(sxy[i, 2] - detector.xy_2[1:J_2,2])^2
p_2[i, 1:J_2] <- p0_2 * exp(-d_squared_2[i,1:J_2])/(2*sigma*sigma)) *
z[i]                                  ## equation (4)
    }
```

```
for (j in 1:J_2) {
pdot_2[j] <- 1 - prod((1 - p_2[1:M, j]))        ## equation (5)
    }
    ydot_2[1:J_2] ~ dbern_vector(pdot_2[1:J_2], 1)    ## equation (7)
 })

Model_4 <- nimbleCode({
   ##-------------------------------------------------------------------
-----------------------------
## AC LOCATIONS
    for (i in 1:M) {                                ## equation (1)
        sxy[i, 1] ~ dunif(0, x.max)
        sxy[i, 2] ~ dunif(0, y.max)
    }

   ## INDIVIDUAL INCLUSION
    psi ~ dunif(0, 1)
    for (i in 1:M) {
        z[i] ~ dbern(psi)                           ## equation (2)
    }

    N <- sum(z[1:M])                                ## equation (3)


    sigma ~ dunif(0, 100)
    p0_1 ~ dunif(0, 1)
    p0_2 ~ dunif(0, 1)
    alpha_1 ~ dunif(0, 1)


## SURVEY TYPE 1: IDENTIFIED + UNIDENTIFIED DETECTIONS
    for (i in 1:M) {
d_squared_1[i, 1:J_1] <- (sxy[i, 1] - detector.xy_1[1:J_1,1])^2 +
(sxy[i, 2] - detector.xy_1[1:J_1,2])^2
p_1[i, 1:J_1] <- p0_1 * exp(-d_squared_1[i,1:J_1])/(2*sigma*sigma))
## equation (4)
##p_1 in model 1 and 3 is equivalent to p_1*alpha in model 2 and 4


y_1[i, 1:J_1] ~ dbern_vector(p_1[i, 1:J_1] * alpha_1 , z[i])
## equation (6)
         punid_1[i, 1:J_1] <- p_1[i, 1:J_1] * (1 - alpha_1) * z[i]
    }
    for (j in 1:J_1) {
        pdot_1[j] <- 1 - prod((1 - punid_1[1:M, j]))
## equation (5)
    }
    ydot_1[1:J_1] ~ dbern_vector(pdot_1[1:J_1], 1)
## equation (7)
```

```
## SUVERY TYPE 2: UNIDENTIFIED DETECTIONS
    for (i in 1:M) {
d_squared_2[i, 1:J_2] <- (sxy[i, 1] - detector.xy_2[1:J_2,1])^2 +
(sxy[i, 2] - detector.xy_2[1:J_2,2])^2
p_2[i, 1:J_2] <- p0_2 * exp(-d_squared_2[i, 1:J_2]/(2*sigma*sigma)) *
z[i]                                ## equation (4)
    }
    for (j in 1:J_2) {
     pdot_2[j] <- 1 - prod((1 – p_2[1:M, j]))    ## equation (5)
    }
    ydot_2[1:J_2] ~ dbern_vector(pdot_2[1:J_2], 1)  ## equation (7)
})
```