Multi session analysis 2019

Fernando Mateos-Gonzalez

This report analises data from capture-recapture sessions during 17-18-19 April, 24-25 April and 15-19 July 2019, in order to calculate the density of the population.

Analysis

This analysis was carried out with the package 'secr' version 3.2.1 and R version 3.6.1

We start by creating the capthist, the file combining our captures with the trap locations:

```
multi <- read.capthist(here("data", "multicaptures.txt"), here("data", "traps
.txt"), detector = "proximity")

## Session 2
## More than one detection per detector per occasion at binary detector(s)

summary(multi)

## $`1`
## Object class       capthist
## Detector type      proximity
## Detector number    20
## Average spacing    13.57938 m
## x-range            -623778.5 -623653.2 m
## y-range            -1187164 -1187091 m
##
## Counts by occasion
##                     1  2  3  4  5 Total
## n                  15 23 15 11 14    78
## u                  15 22 11  7 11    66
## f                  56  8  2  0  0    66
## M(t+1)             15 37 48 55 66    66
## losses              0  0  0  0  0     0
## detections         15 23 15 11 14    78
## detectors visited  8 13  9 10 10    50
## detectors used    20 20 20 20 20   100
##
## $`2`
## Object class       capthist
## Detector type      proximity
## Detector number    20
## Average spacing    13.57938 m
## x-range            -623778.5 -623653.2 m
## y-range            -1187164 -1187091 m
##
```

```
## Counts by occasion
##                     1  2  3  4  5 Total
## n                  23 27 16 19  8    93
## u                  23 22 12 11  5    73
## f                  58 10  5  0  0    73
## M(t+1)             23 45 57 68 73    73
## losses              0  0  0  0  0     0
## detections         24 27 17 21  8    97
## detectors visited 14 12 12 15  8    61
## detectors used    20 20 20 20 20   100
```

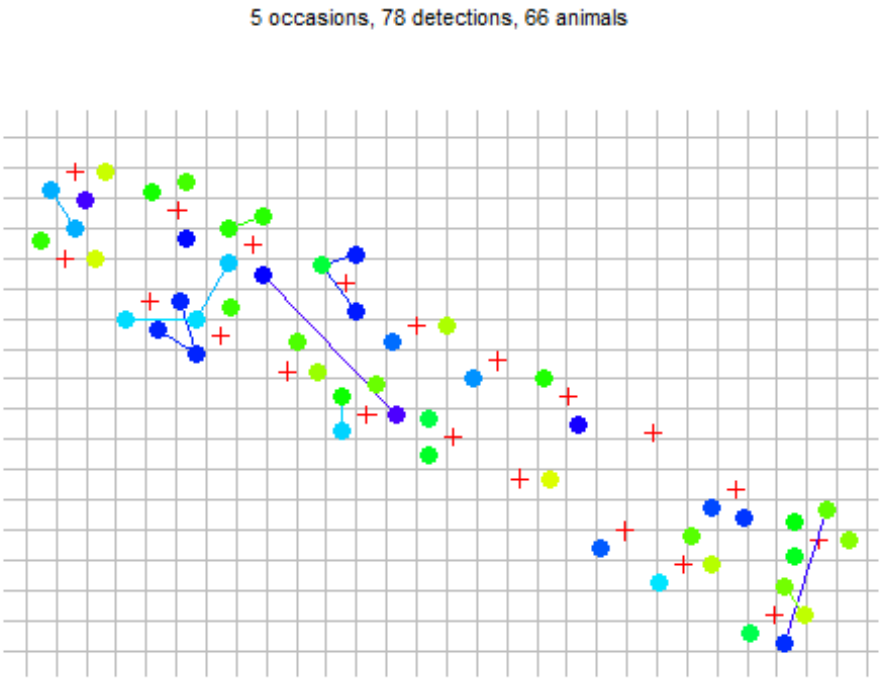
```
str(multi)
```

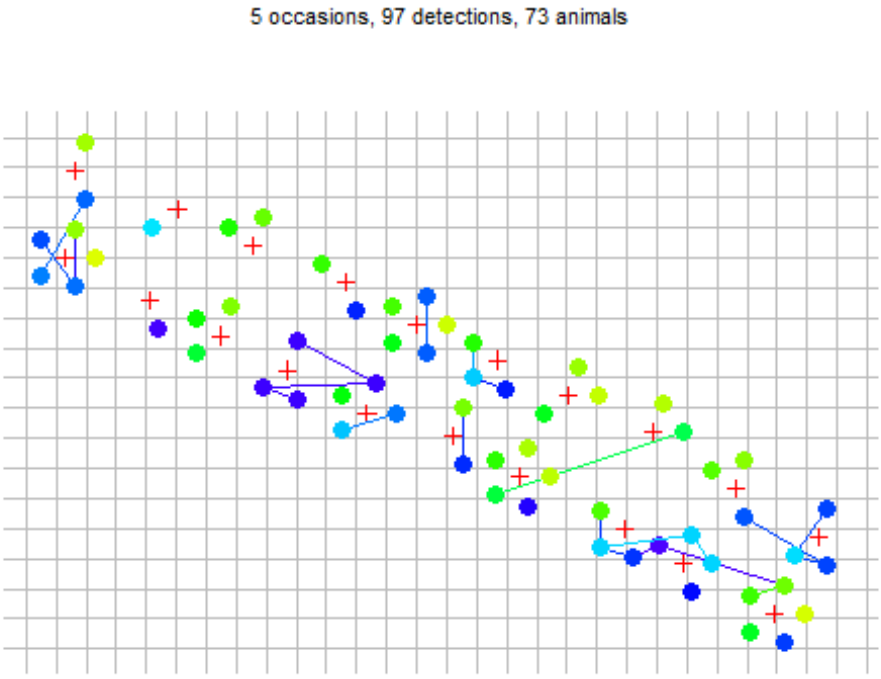
```
## List of 2
##  $ 1: 'capthist' num [1:66, 1:5, 1:20] 0 1 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 3
##   .. ..$ : chr [1:66] "2019041701" "2019041702" "2019041703" "2019041704"
...
##   .. ..$ : chr [1:5] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:20] "1" "2" "3" "4" ...
##   ..- attr(*, "covariates")='data.frame':    0 obs. of  0 variables
##   ..- attr(*, "traps")=Classes 'traps' and 'data.frame': 20 obs. of  2 var
iables:
##   .. ..$ x: num [1:20] -623777 -623760 -623747 -623732 -623720 ...
##   .. ..$ y: num [1:20] -1187091 -1187097 -1187103 -1187109 -1187116 ...
##   .. ..- attr(*, "detector")= chr "proximity"
##   .. ..- attr(*, "spacex")= num 1.61
##   .. ..- attr(*, "spacey")= num 0.743
##   .. ..- attr(*, "spacing")= num 13.6
##   ..- attr(*, "session")= chr "1"
##  $ 2: 'capthist' num [1:73, 1:5, 1:20] 0 0 0 0 0 0 0 0 0 1 ...
##   ..- attr(*, "dimnames")=List of 3
##   .. ..$ : chr [1:73] "2019071501" "2019071502" "2019071503" "2019071504"
...
##   .. ..$ : chr [1:5] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:20] "1" "2" "3" "4" ...
##   ..- attr(*, "covariates")='data.frame':    0 obs. of  0 variables
##   ..- attr(*, "traps")=Classes 'traps' and 'data.frame': 20 obs. of  2 var
iables:
##   .. ..$ x: num [1:20] -623777 -623760 -623747 -623732 -623720 ...
##   .. ..$ y: num [1:20] -1187091 -1187097 -1187103 -1187109 -1187116 ...
##   .. ..- attr(*, "detector")= chr "proximity"
##   .. ..- attr(*, "spacex")= num 1.61
##   .. ..- attr(*, "spacey")= num 0.743
##   .. ..- attr(*, "spacing")= num 13.6
##   ..- attr(*, "session")= chr "2"
##  - attr(*, "class")= chr [1:2] "capthist" "list"
##  - attr(*, "inject.time")= num [1:175] 0 0 0 0 0 0 0 0 0 0 ...
```

n number of distinct individuals detected on each occasion t
u number of individuals detected for the first time on each occasion t
f number of individuals detected on exactly t occasions
M(t+1) cumulative number of detected individuals on each occasion t

Now we use the plot method, which for capthist objects has additional arguments; we set tracks = TRUE to join consecutive captures of each individual.

```
par(mfrow = c(1,1), mar = c(1,1,1,1)) # reduce margins
plot (multi, tracks = TRUE, gridsp = 5,border = 10)
```

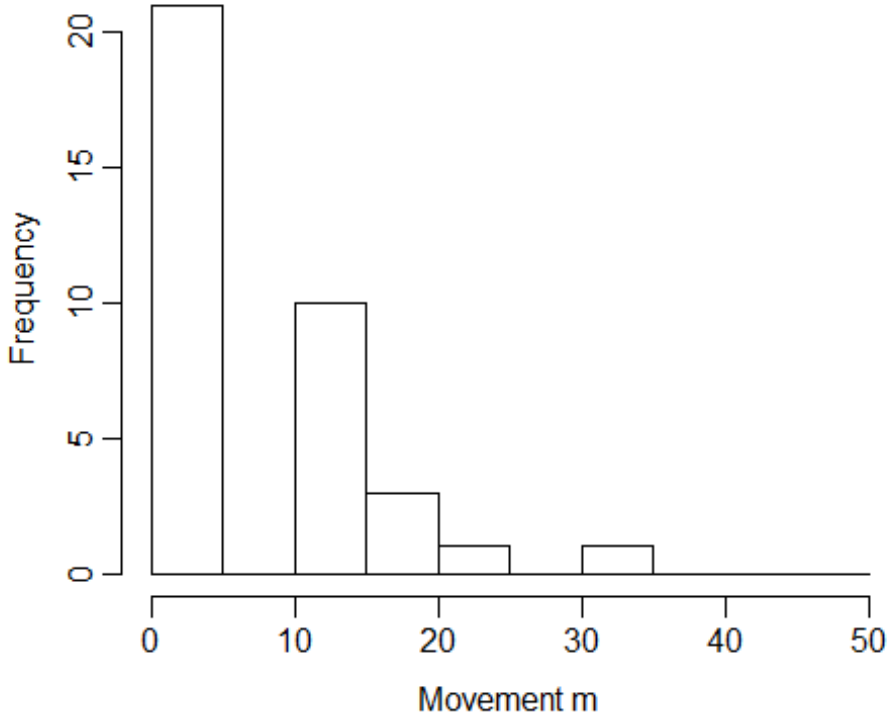




The most important insight from this figure is that individuals tend to be recaptured near their site of first capture. This is expected when the individuals of a species occupy home ranges. In SECR models the tendency for detections to be localised is reflected in the spatial scale parameter σ.

Successive trap-revealed movements can be extracted with the moves function and summarised with hist:

```
m <- unlist(moves(multi))
par(mar = c(3.2,4,1,1), mgp = c(2.1,0.6,0)) # reduce margins
hist(m, breaks = seq(0/5, 50,5), xlab = "Movement m", main = "")
```



We will employ the estimate of the spatial scale σ from the July session (6.21) to fit the simplest possible SECR model with function secr.fit.

```
fit <- secr.fit (multi, buffer = 4 * 6.21, trace = FALSE,biasLimit = NA, veri
fy = FALSE)

detector(traps(multi)) <- "proximity"

fit

##
## secr.fit(capthist = multi, buffer = 4 * 6.21, verify = FALSE,
##     biasLimit = NA, trace = FALSE)
## secr 3.2.1, 17:31:54 23 Jul 2019
##
## $`1`
## Detector type      proximity
## Detector number    20
## Average spacing    13.57938 m
## x-range            -623778.5 -623653.2 m
## y-range            -1187164 -1187091 m
##
## $`2`
## Detector type      proximity
## Detector number    20
## Average spacing    13.57938 m
## x-range            -623778.5 -623653.2 m
```

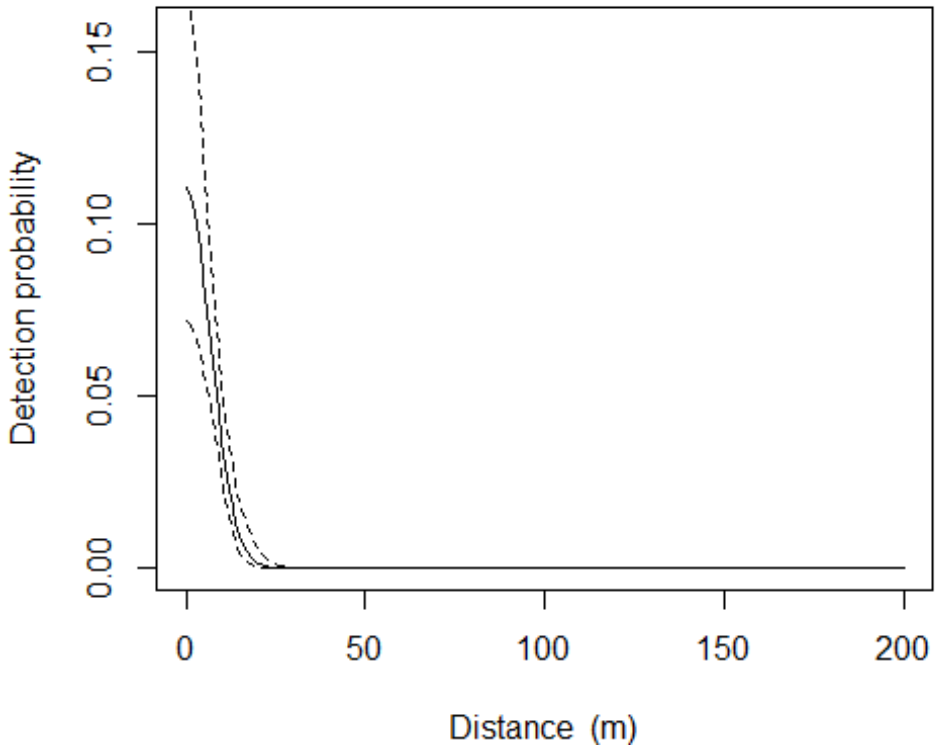
```
## y-range            -1187164 -1187091 m
##
##
##             1  2
## Occasions   5  5
## Detections 78 97
## Animals    66 73
## Detectors  20 20
##
## Model           :  D~1 g0~1 sigma~1
## Fixed (real)    :  none
## Detection fn    :  halfnormal
## Distribution    :  poisson
## N parameters    :  3
## Log likelihood  :  -366.486
## AIC             :  738.972
## AICc            :  739.1498
##
## Beta parameters (coefficients)
##            beta    SE.beta       lcl       ucl
## D      5.619811 0.16284346  5.300643  5.938978
## g0    -2.087030 0.24087362 -2.559134 -1.614926
## sigma  1.912911 0.09553052  1.725675  2.100148
##
## Variance-covariance matrix of beta parameters
##                 D          g0        sigma
## D      0.02651799 -0.01606428 -0.004975350
## g0    -0.01606428  0.05802010 -0.014584702
## sigma -0.00497535 -0.01458470  0.009126079
##
## Fitted (real) parameters evaluated at base levels of covariates
##
##  session = 1
##        link    estimate SE.estimate          lcl         ucl
## D       log 275.8371276 45.21771018 200.46571867 379.5467946
## g0    logit   0.1103638  0.02364985   0.07181526   0.1659058
## sigma   log   6.7727783  0.64848597   5.61631085   8.1673766
##
##  session = 2
##        link    estimate SE.estimate          lcl         ucl
## D       log 275.8371276 45.21771018 200.46571867 379.5467946
## g0    logit   0.1103638  0.02364985   0.07181526   0.1659058
## sigma   log   6.7727783  0.64848597   5.61631085   8.1673766
```

The report comprises:
• function call and time stamp
• summary of the data
• description of the model, including the maximized log likelihood, Akaike's Information Criterion AIC

• estimates of model coefficients (beta parameters)
• estimates of variance-covariance matrix of the coefficients
• estimates of the 'real' parameters

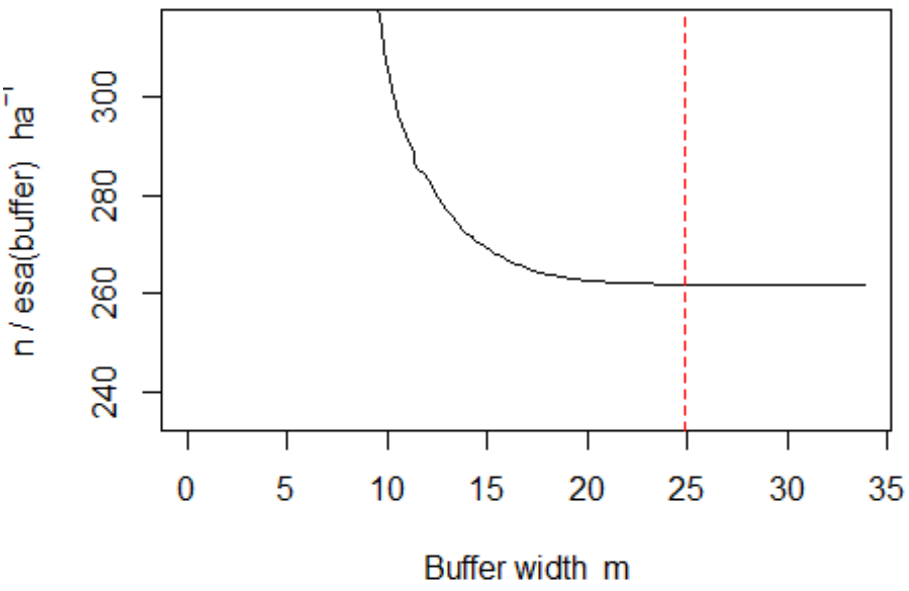
. The estimated density is 275 susliks per hectare, 95% confidence interval 200-379 susliks per hectare
. The other two real parameters jointly determine the detection function, plotted below with 95% confidence limits

```
par(mar = c(4,4,1,1)) # reduce margins
plot(fit, limits = TRUE)
```



The theory of SECR tells us that buffer width is not critical as long as it is wide enough that animals at the edge have effectively zero chance of appearing in our sample.We check that for the present model with the function esa.plot.
The estimated density has easily reached a plateau at the chosen buffer width (dashed red line):

```
esa.plot(fit)
abline(v = 4 * 6.21, lty = 2, col = 'red')
```


n / esa(buffer) ha⁻¹
240
260
280
300
0
5
10
15
20
25
30
35
Buffer width m

```
```