Home ranges

Animal tracking 25/26

Giuliano Colosimo

University of Rome Tor Vergata

Table of contents I

What is a home range?

What is a home range?

- Home range estimators are widely used in spatial ecology studies, as they provide basic measurements of animal space-use patterns
- Home range estimators are a critical component for understanding animal spatial ecology. The choice of home range estimator in spatial ecology studies can significantly influence management and conservation actions, as different methods lead to vastly different inter- pretations of movement patterns, habitat selection, as well as home range requirements (Silva, Crane, Suwanwaree, Strine, & Goode, 2018)

- The geography area to which an organism normally confines its activities
- The extent of space animals use to live and reproduce (Viana et al., 2018)
- Home range is defined by the interaction between animals and the environment, and its size is the direct result of movement driven by habitat selection and other external factors, biotic interactions, and intrinsic factors related to individual state and characteristics (Börger, Dalziel, & Fryxell, 2008)
- Home range formation is thus the result of dynamic processes. Both the habitat and internal state of animals might change through time and cause home range size to vary (Viana et al., 2018)

Estimates of home-range

Estimates of home-range

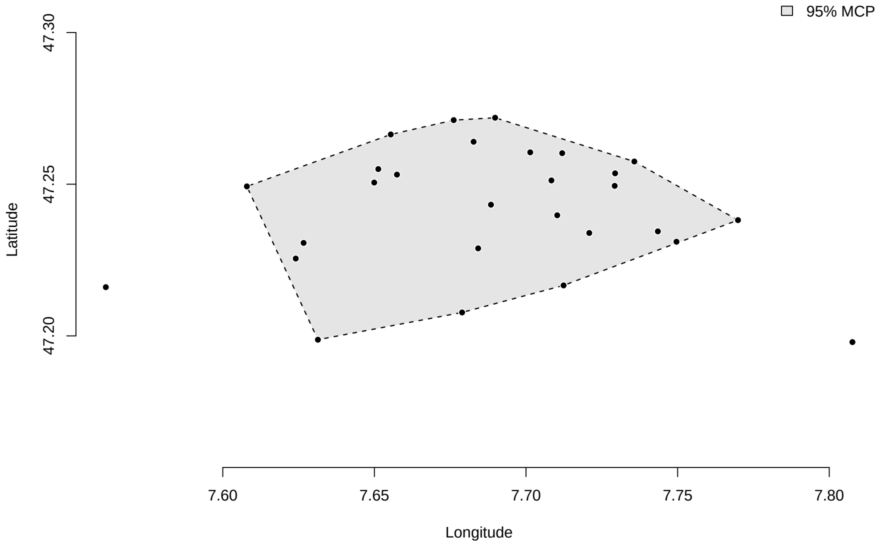
- Minimum Convex Polygon
  - The simplest way to draw the boundaries of a home range from a set of location data is to construct the smallest possible convex polygon around the data
- Kernel Density Estimators
  - Extrapolate where a geo-tracked animal spends its time
- Kriging
  - Spatial prediction based on a geostatistical model (Webster, 2007)
- Brownian Bridge Movements
  - Explicitly models an animal's movement path and estimates an animal mobility feature called Brownian motion variance

Minimum Convex Polygon

Minimum Convex Polygon

```
library(move)
library(adehabitatHR)
library(scales)
bats <-
  move("PATH/TO/Parti-colored bat Safi Switzerland.csv")
X330 <- bats[["X330"]]
X330$id <- "X330"
mcpX330<-mcp(as(X330[,'id'], 'SpatialPointsDataFrame'))
plot(X330, type="n", bty="na", xlab="Longitude",
     ylab="Latitude")
plot(mcpX330, col="grey90", lty=2, lwd=1.25, add=TRUE)
points(X330, pch=16)
points(X330, pch=1, col="white")
legend("topright", as.character("95% MCP"),
       fill="grey90", bty="n")
```

Estimation using MCP



- The function to estimate MCP is the following:

```
mcp(xy, percent=95, unin = c("m", "km"),
    unout = c("ha", "km2", "m2"))
```

- The argument *percent* in the function determines the percentage of data points that will be considered in the estimate
  - A single number for the function mcp and a vector for the function mcp.area: 100 minus the proportion of outliers to be excluded from the computation.
- Remember: you can always access details on a specific function by typing:

```
?NAME_OF_FUNCTION
```

- Lets take a look at the estimated polygon

```
mcpX330
```

```
Object of class "SpatialPolygonsDataFrame" (package sp):

Number of SpatialPolygons:  1

Variables measured:
       id         area
X330 X330 7.053224e-07
```

- Area value seems strange!!!

Question (?)

What is happening here? Can you guess what the issue may be?

- Area value seems strange!!!

! Question (?)

What is happening here? Can you guess what the issue may be?

- *adehabitatHR* calculated the area according to the units of the projection (geographic coordinates system based on long/lat), in this case decimal degrees
- Therefore we have to project our data into a different coordinate system

! Question (?)

What coordinate system should we use?

! Question (?)

What coordinate system should we use?

- Should be a projected CRS
- Could be a general one or, since we are restricted to Switzerland, we it could be something focused on the Swiss territory
- We can go to the EPSG website and find out what works for us

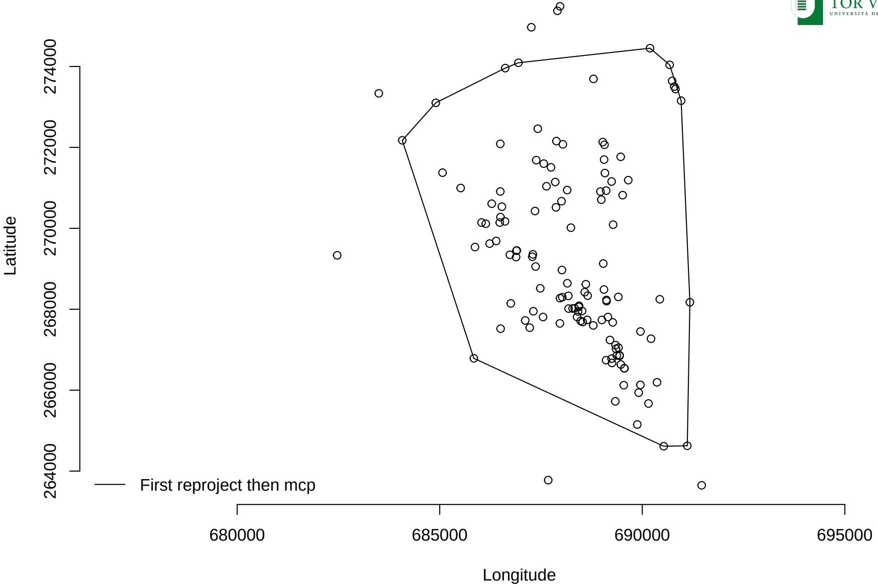
Figure 1: Write “Switzerland” in the search bar and lets see what comes out!

- We select the EPSG:21781 and use those parameters

- We select the EPSG:21781 and use those parameters
- Now we can transform the CRS for all individuals and we can calculate the MCP again

```
bats$id <- trackId(bats)
mcpData<-mcp(as(bats[,'id'],'SpatialPointsDataFrame'))
bats.proj <- spTransform(bats,
    CRS("+proj=somerc +lat_0=46.95240555555556
        +lon_0=7.439583333333333 +x_0=600000 +y_0=200000
        +ellps=bessel +units=m +no_defs"))
mcpData.proj <- mcp(as(bats.proj[, 'id'],
                        'SpatialPointsDataFrame'))
```

```r
plot(bats.proj[["X21"]], bty="na",
     xlab="Longitude", ylab="Latitude")
plot(mcpData.proj[mcpData.proj$id=="X21",], add=TRUE)
legend("bottomleft", c("First reproject then mcp"),
       lty=1, bty="n")
```


Latitude
274000
272000
270000
268000
266000
264000
First reproject then mcp
680000
685000
690000
695000
Longitude

Size of MCP changes with sampling effort

```
hrBootstrap(bats[['X21']], rep=500, levelMax=95)
legend("bottomright",
legend=c("real MCP size","100% percentil",
         "75% percentil","50% percentil",
         "25% percentil","0% percentil"),
lty=c(4,2,3,1,3,2), col=c("black","cyan","red",
                          "black","red","cyan"))
```

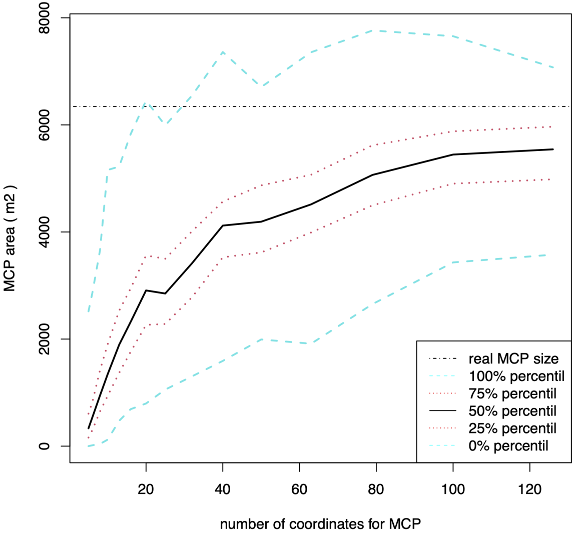
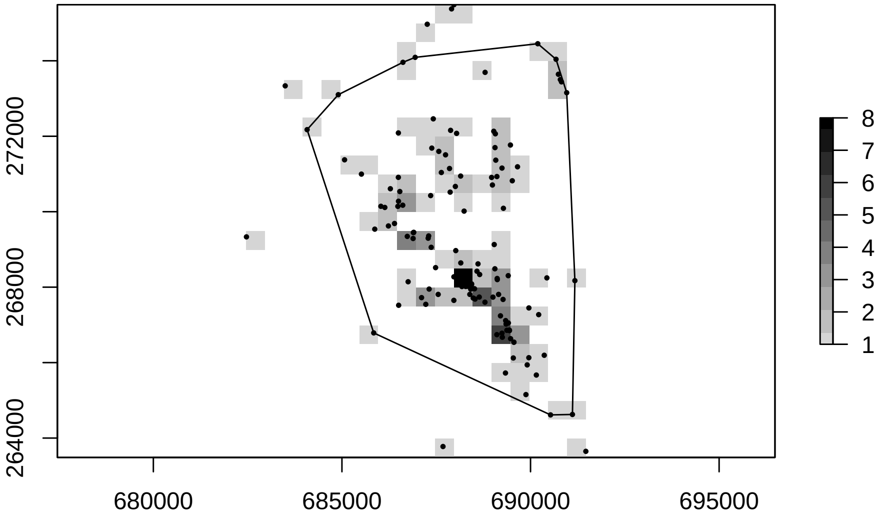


Figure 2: Bootstrap of MCP

KDE

KDE

```r
template <- raster(extent(bats.proj[[1]]))
res(template)<-500
count <- rasterize(split(bats.proj)[[1]],
                   template,field=1,  fun="count")
plot(count, col=grey(10:0/12))
plot(mcpData.proj[1,], add=TRUE)
points(bats.proj[[1]], pch=16, cex=0.5)
```

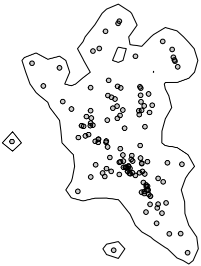

272000
268000
264000
680000
685000
690000
695000
8
7
6
5
4
3
2
1

- h: degree of smoothness or how tightly the data should be hugged by the distribution function
  - h="LSCV": h calculated from the data via least square cross validation
  - h="ad-hoc": h calculated from the data via sample size and spatial spread

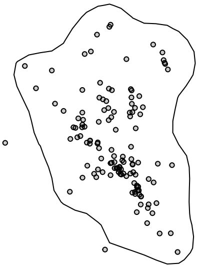
```
X21 <- bats.proj[['X21']]
kern1 <- kernelUD(as(X21, "SpatialPoints"), h=500)
kern2 <- kernelUD(as(X21, "SpatialPoints"))
kern3 <- kernelUD(as(X21, "SpatialPoints"), h=2000)
kern4 <- kernelUD(as(X21, "SpatialPoints"), h="LSCV")
kern <- c("kern1", "kern2", "kern3", "kern4")
hName <- c("h=500",
           "h='ad-hoc'",
           "h=2000",
           "h=LSCV")
```

```r
par(mfrow=c(2,2))
par(mar=c(1,0.5,3,0.5))
for(i in 1:4){
  plot(getverticeshr(get(kern[i])))
  points(X21, pch=16, cex=0.75, col=alpha("black", 0.2))
  points(X21, cex=0.75)
  title(hName[i])
}
```

h=500



h='ad−hoc'



h=2000

h=LSCV

References

References I

Börger, L., Dalziel, B. D., & Fryxell, J. M. (2008). Are there general mechanisms of animal home range behaviour? A review and prospects for future research. *Ecology Letters*, *11*(6), 637–650. Wiley.

Silva, I., Crane, M., Suwanwaree, P., Strine, C., & Goode, M. (2018). Using dynamic brownian bridge movement models to identify home range size and movement patterns in king cobras. (U. G. Munderloh, Ed.)*PLOS ONE*, *13*(9), e0203449. Public Library of Science (PLoS).

References II

Viana, D. S., Granados, J. E., Fandos, P., Pérez, J. M., Cano-Manuel, F. J., Burón, D., Fandos, G., et al. (2018). Linking seasonal home range size with habitat selection and movement in a mountain ungulate. *Movement Ecology*, 6(1). Springer Science; Business Media LLC. Retrieved from https://doi.org/10.1186%2Fs40462-017-0119-8

Webster, R. (2007). *Geostatistics for environmental scientists*. (M. Oliver, Ed.)Statistics in practice (2nd ed.). Chichester: Wiley.