

Appendix 1: Home Range Estimation using Autocorrelated Kernel Density Estimation

Odd Jacobson

April 18, 2023

The purpose of this appendix is to detail the steps to estimating a home range (HR) using continuous-time movement modelling and the *ctmm* package. This document can be used as a practical guide, where one can use our practice dataset or one's own data to walk-through the analytical process.

Accounting for autocorrelation is important so that we avoid biases in our results. However, it requires some additional steps compared to most conventional estimators, which is why we describe the process and provide an example workflow. We strongly recommend going through the *ctmm* vignettes (see <https://ctmm-initiative.github.io/ctmm/index.html>) for a more detailed review.

Generating a home range estimate from movement data using continuous-time movement modelling involves three main steps: 1) variogram inspection, 2) model fitting and selection, and 3) Autocorrelated Kernel Density Estimation (AKDE). This process can either be done using the *ctmm* package in the R environment for statistical computing (R Core Team 2022), or using the *ctmmweb* point-and-click graphical user interface (Calabrese et al. 2021), which streamlines the modelling steps, helping users conduct home range analysis without the need to know the R programming language. We describe the process using R below:

The first step is to load the necessary packages and prepare the data

```
library(tidyverse)
library(ctmm)
```

Prepare Data

The data must have the same format as the following dataframe with the same column names. These are the same format required by *Movebank*. Either you can manually edit the dataframe and then convert to a telemetry object, or put data on Movebank and import from there, which will automatically put the data in the correct format.

Note: `individual.local.identifier` (ILI) specifies the unique ID (usually individual or group) that you want the home range estimate for. At the bottom of the document, we also include some example code of how to do all of the below analysis in a single step for a list of several ILIs.

In our study, the ILI indicated the different sampling regimes. "All" was the ILI for the complete segments. For this walkthrough, we will use the data from the complete segment of SP group, and change the ILI to the group.

```
# read in data frame
# filtering data from SP group, and taking the data from the complete segment - denoted
  ↪ as "all"
DATA <- read.csv("Data/CH1_GPS_data.csv", row.names = NULL) %>%
  filter(group == "SP" & individual.local.identifier == "all") %>%    #select the complete
  ↪ segment from SP group (could pick any group)
```

```
dplyr::select(-individual.local.identifier) %>% # remove prev ILI column
rename(individual.local.identifier = group) # make group the new ILI
```

The data should look like this (these are the first six rows):

```
head(DATA)
```

```
##   individual.local.identifier      timestamp location.long location.lat
## 1                        SP 2010-09-12 05:00:00      -85.37901      10.50080
## 2                        SP 2010-09-12 05:30:00      -85.37907      10.50041
## 3                        SP 2010-09-12 06:00:00      -85.37932      10.49956
## 4                        SP 2010-09-12 06:30:00      -85.37947      10.49877
## 5                        SP 2010-09-12 07:00:00      -85.37920      10.49790
## 6                        SP 2010-09-12 07:30:00      -85.37858      10.49784
```

Once the dataframe is in correct format, convert it to a telemetry object and specify the UTM projection:

```
# convert data to telemetry object
DATA <- DATA %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↪ +ellps=WGS84 +towgs84=0,0,0")
```

Plot the data:

```
# plot location data
plot(DATA, main = "Location Data")
```

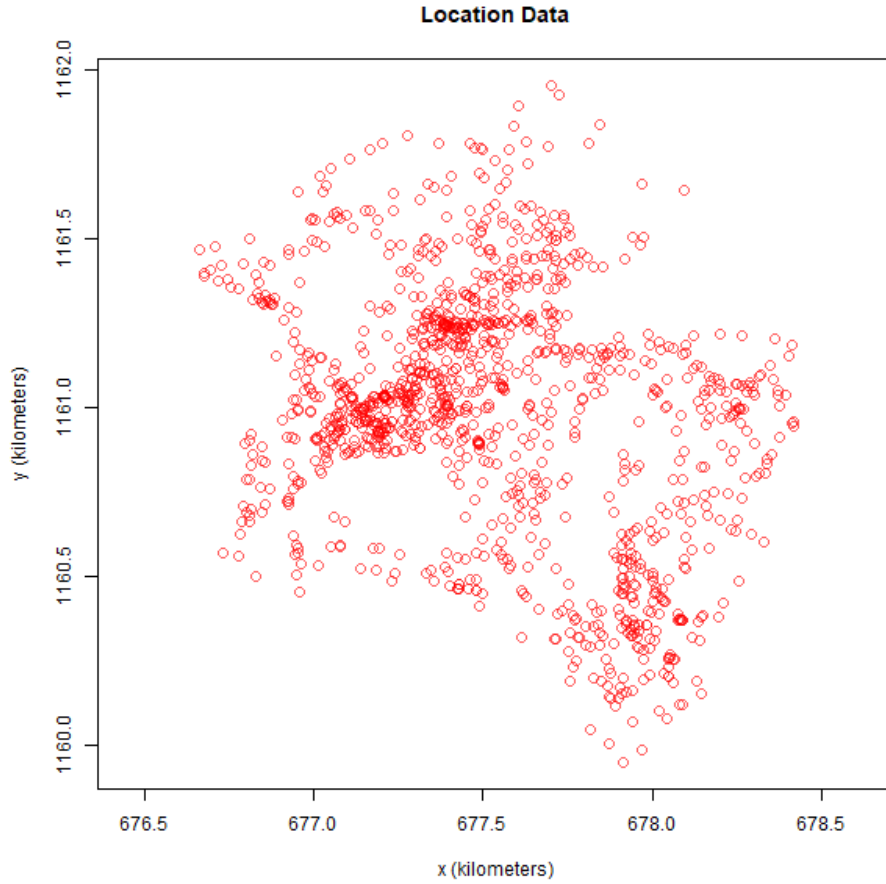


Figure 1: Location data from the complete segment of SP group

Variogram Inspection

Variograms plot the semi-variance (y-axis), which is a measure of the average squared displacement, as a function of the time-lag that separates any pair of observed locations (Diggle and Ribeiro 2007; Silva et al. 2021). Variograms play two major roles in the *ctmm* workflow: first, they provide an unbiased visual diagnostic to assess the autocorrelation structure present in the data, and second, they inform whether the data shows evidence of range residency (Silva et al. 2021). Asymptoting curves in a variogram indicate range residency. Where the asymptote aligns with the x-axis is a measure of the necessary time-lag between positions to assume independence (Silva et al. 2021). It is also a rough estimate of the home range crossing time (Christen H. Fleming and Calabrese 2017). If the curve continues to increase without flattening, the animals are either non-resident (i.e. home range drift or migration), or not tracked long enough to capture the full extent of their home ranges (Calabrese, Fleming, and Gurarie 2016).

We used the *variogram* and *plot* functions to visually assess the empirical variogram curves. Once the data are confirmed to represent range-restricted movement, we can proceed with model fitting and selection. It is necessary to confirm range-residency before conducting home range estimation because, while *ctmm* is capable of fitting both range-resident (the default) and endlessly diffusing movement models, only the first set are appropriate for home range estimation.

```
SVF <- variogram(DATA, dt = c(1,10) %%% "hour") # dt argument changes the width of the
  ↪ time-lag bins (makes variogram smoother)
```

```
plot(SVF, main = "Empirical Variogram")
```

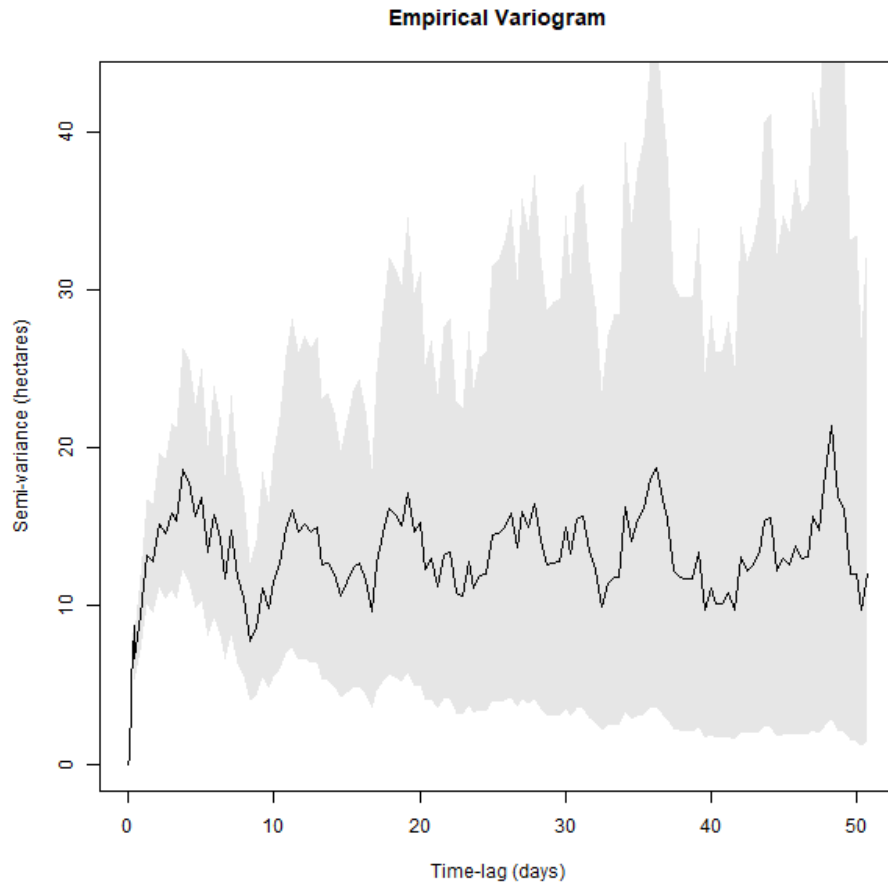


Figure 2: Empirical variogram from the complete segment of SP group

This is a plot of the empirical variogram. The line asymptotes at approximately three days. This is roughly the average home range crossing time. This is also approximately how far apart locations need to be in time for them to be independent (<https://ctmm-initiative.github.io/ctmm/articles/variogram.html>).

Movement Model Selection

Model fitting also involves two steps: first the `ctmm.guess` function uses the shape of the empirical variogram to generate starting values required for the non-linear models, and second, the `ctmm.select` function uses the values calculated from `ctmm.guess` to fit a range of alternative stationary (and range-restricted) movement models using Maximum Likelihood (Christen H. Fleming et al. 2014). Models are ranked by AICc (Akaike information criterion) allowing us to evaluate which model or models best predict the data. This process permits identification and fit of a stationary movement model that corresponds to the observed movement behavior of the animal (Christen H. Fleming et al. 2014).

It is worth noting that here, stationary means that the underlying movement processes are assumed to be consistent throughout the duration of the data. Movement model parameters represent time-averaged values, which has important implications on how data should be segmented for home range analysis. If

the underlying parameters change drastically within the sample—particularly the mean location—then the stationary assumption has been violated. Therefore, it is common practice to segment the data when the parameters change and estimate separate ranges. This is consistent with Burt’s original concept of the home range where, for example, he stated winter and summer ranges for migratory species should be considered separately with the travel between as transit (Burt 1943). In our case, all sampling regimes were from single, stationary ranges which negated any need for further segmentation.

The pool of potential movement models which involve home range behavior include:

- 1) *Independent and Identically Distributed* (IID) – location data has uncorrelated positions and velocities.
- 2) *Ornstein-Uhlenbeck* (OU) – location data has autocorrelated positions and uncorrelated velocities.
- 3) *OU Foraging* (OUF) – location data has autocorrelated positions and velocities (Calabrese, Fleming, and Gurarie 2016; C. Fleming et al. 2014; Christen H. Fleming et al. 2014)

OU and OUF can be further specified with isotropic or anisotropic versions of each. Isotropic means diffusion is equal on every extent of the home range, while anisotropic means diffusion is asymmetrical (Silva et al. 2021).

Endlessly diffusing movement models (non-HR models) such as *brownian motion* (BM) or *integrated OU* (IOU) cannot be statistically compared to HR models using maximum likelihood (see `?ctmm.select`). To fit these movement models, one must manually specify them.

```
# get starting values for models
GUESS <- ctmm.guess(DATA, interactive=FALSE, variogram = SVF)

# fit models and select top one, trace = 2 allows you to see progress
FIT <- ctmm.select(DATA, GUESS, trace=2)

# see model summary for top model
summary(FIT)
```

```
## $name
## [1] "OUF anisotropic"
##
## $DOF
##      mean      area  diffusion    speed
## 49.56111  81.16699 346.58408 1183.63214
##
## $CI
##              low      est      high
## area (square kilometers)  2.054622  2.586565  3.178821
## [position] (hours)      10.561126 13.624460 17.576337
## [velocity] (minutes)    21.251008 23.651631 26.323441
## speed (kilometers/day)   5.282984  5.437900  5.592755
## diffusion (hectares/day) 39.214897 43.694482 48.412835
```

The top model selected for our practice dataset was *OUF anisotropic*. Above is the summary information for that model. The `$DOF` specifies the effective sample sizes. The most important one for home range estimation being under `area` which indicates the number of statistically independent points (or approximately the number of home range crossings – see Methods in the main text).

The `area (square kilometers)` slot indicates the Gaussian area, which is an estimate of spatial variance, but is not our AKDE area. `tau[position] (hours)` is the tau referenced in the main text. This is time

necessary between locations for them to be independent, or approximately the home range crossing timescale. `tau[velocity]` (minutes) is the timescale necessary for the velocities to be independent. Estimates of speed (i.e. proportional to average daily travel distance) and diffusion rate are also included.

```
# plot empirical variogram with best model
plot(SVF, CTMM = FIT, main = "Variogram and Fitted Model")
```

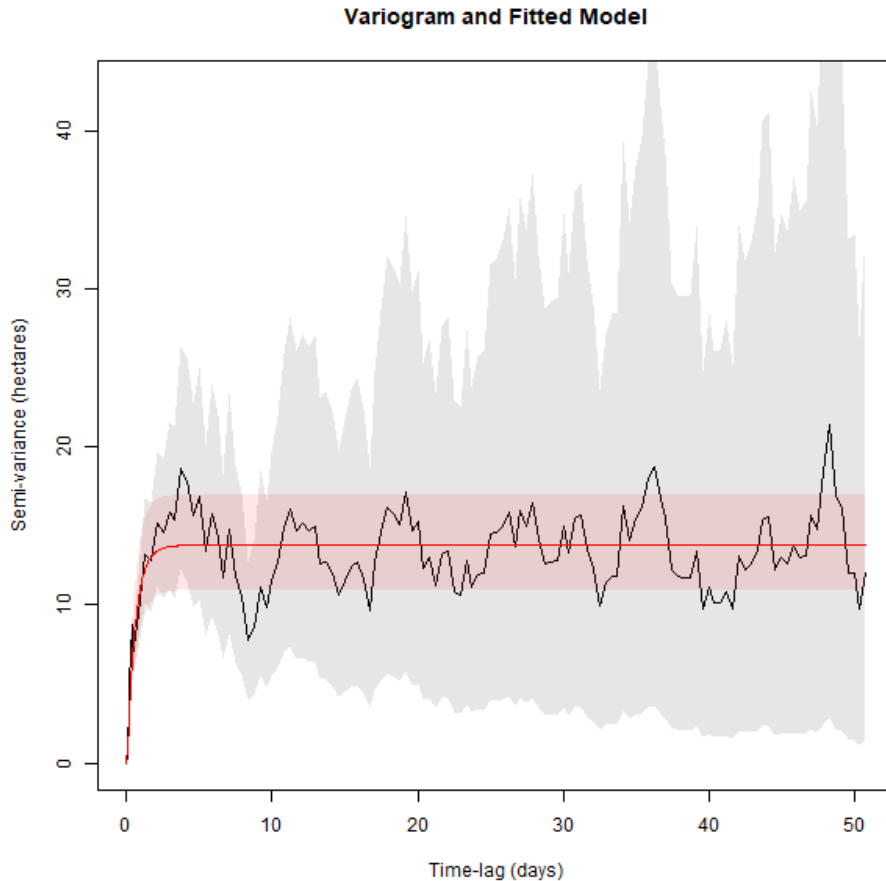


Figure 3: Top movement model fitted to the empirical variogram

AKDE Home Range Estimation

The final step is to calculate an autocorrelated kernel density home-range estimate (AKDE) using the eponymously named `akde` function (Calabrese, Fleming, and Gurarie 2016). This function takes the movement data and the corresponding fitted model and returns: a utilization distribution (UD) object corresponding to the range distribution, information on the optimal bandwidth, point estimates and confidence intervals for HR area, and a measure of the effective sample size of the data for home range estimation. For the sampling regimes in our study, we also included the `weights = TRUE` option, which helps correct for irregular and missing data by down-weighting over-sampled portions of the data and up-weighting under-sampled portions (C. H. Fleming et al. 2018). This helps to offset sampling bias, but is not sufficient if large portions of the true range are missing from the sampled data.

```
# get UD using AKDE
UD <- akde(DATA, FIT, weights = TRUE)

# plot UD over location data
plot(DATA, UD = UD, main = "AKDE Home Range Estimate")
```

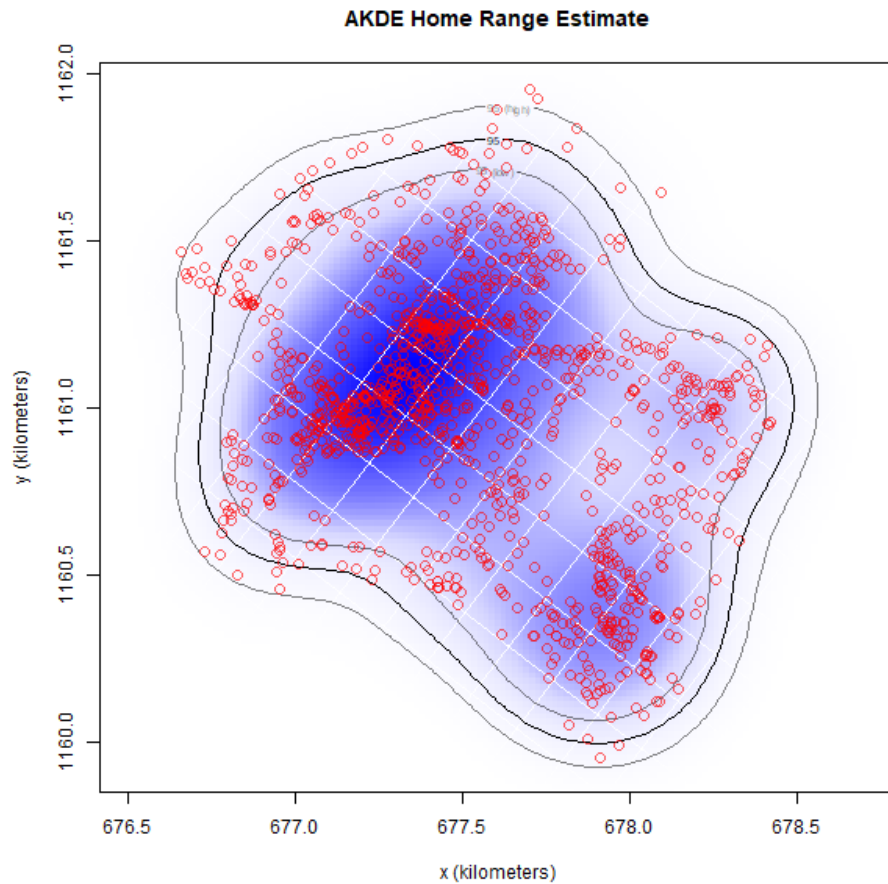


Figure 4: Home range estimate (95% utilization distribution) mean contour and 95% confidence intervals plotted over location data

Below is the summary information:

```
summary(UD)

## $DOF
##      area bandwidth
## 81.16699 145.73448
##
## $CI
##              low      est      high
## area (square kilometers) 1.745174 2.197001 2.700057
##
```

```
## attr("class")
## [1] "area"
```

The effective sample size (DOF area) is the same as from the fitted model. In this case, there was about 81 observed home range crossings in the data. The `area` (square kilometers) slot shows the estimated home range area and 95% confidence intervals.

Example of bulk calculation for several individuals or groups

Below is code to demonstrate how to perform the above analysis with multiple individuals or groups in one step using a loop (may take a little while to run, ~20min).

```
## PREPARE DATA
# take only complete segments (all)
# make group the new individual.local.identifier (ILI)
# change to tele object
DATA_bulk <- read.csv("Data/CH1_GPS_data.csv", row.names = NULL) %>%
  filter(individual.local.identifier == "all") %>% # take only complete segments
  dplyr::select(-individual.local.identifier) %>% # remove prev ILI column
  rename(individual.local.identifier = group) %>% # make group the new ILI
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↪ +ellps=WGS84 +towgs84=0,0,0")

# note: when there are multiple individual.local.identifiers, as.telemetry makes a list,
  ↪ with each
# individual.local.identifier being an element in the list
# variograms, model fits, and UDs follow the same list format

# make empty lists to be filled by below loop
UDs <- FITs <- SVFs <- list()

## BULK CALCULATIONS
# for every ILI, make a variogram (SVF), get starter values (GUESS), select model (FIT),
  ↪ and calculate AKDE (UD)
# grid argument in akde aligns UDs so that overlap function is possible if desired later
  ↪ on
for(i in 1:length(DATA_bulk)){
  SVFs[[i]] <- variogram(DATA_bulk[[i]])
  GUESS <- ctm.guess(DATA_bulk[[i]], interactive=FALSE, variogram = SVFs[[i]])
  FITs[[i]] <- ctm.select(DATA_bulk[[i]], GUESS, trace=2)
  UDs[[i]] <- akde(DATA_bulk[[i]], FITs[[i]], weights = TRUE,
    ↪ grid=list(dr=10, align.to.origin=TRUE))
}

# make names of variograms, fits, and UDs the same as data
names(UDs) <- names(FITs) <- names(SVFs) <- names(DATA_bulk)

# make color blind palette
colorblind_pal <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#D55E00", "#CC79A7")

# plot together
plot(UDs, col.DF=colorblind_pal)
```

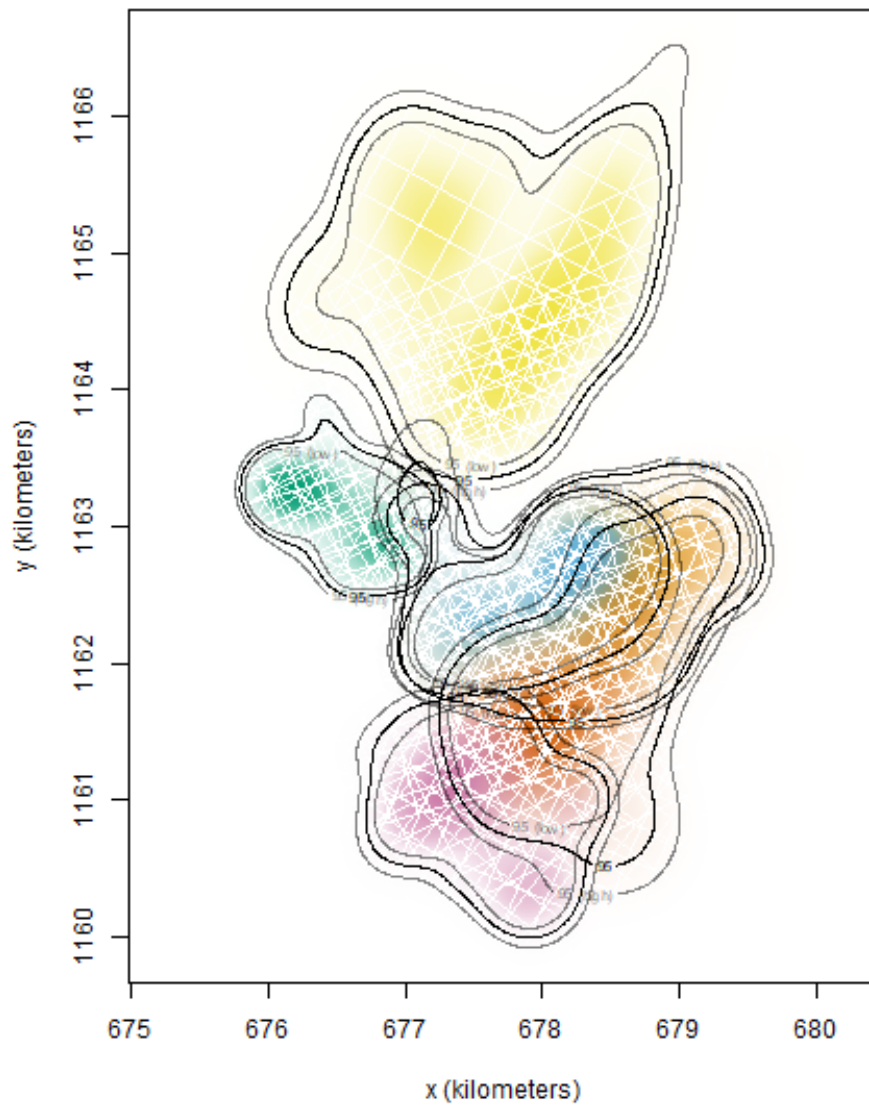



Figure 5: Home range estimate (95% utilization distribution) mean contours and 95% confidence intervals from the six complete segments

```
# plot seperate
par(mfrow = c(2,3))
for(i in 1:length(UDs)){
  plot(DATA_bulk[[i]], UDs[[i]],
       col = colorblind_pal[[i]],
       col.DF=colorblind_pal[[i]],
       main = names(UDs[i]))
}
```

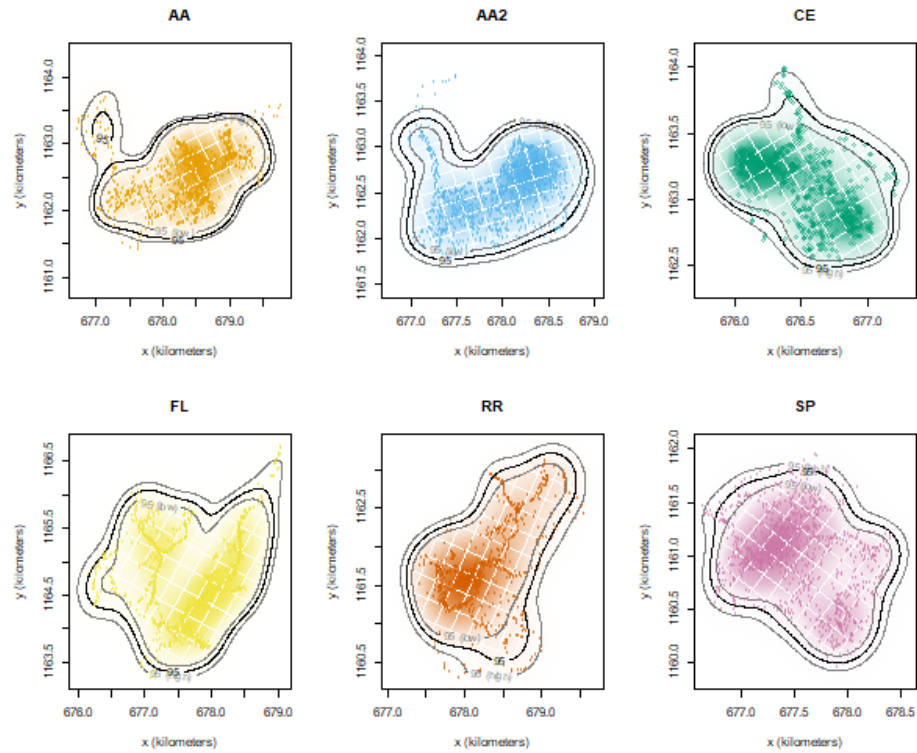


Figure 6: Home range estimate (95% utilization distribution) contours and 95% confidence intervals from the six complete segments plotted separately

You can compare home range areas using:

```
meta(UDs, variable = "area", main = "HR areas" col = colorblind_pal)
```

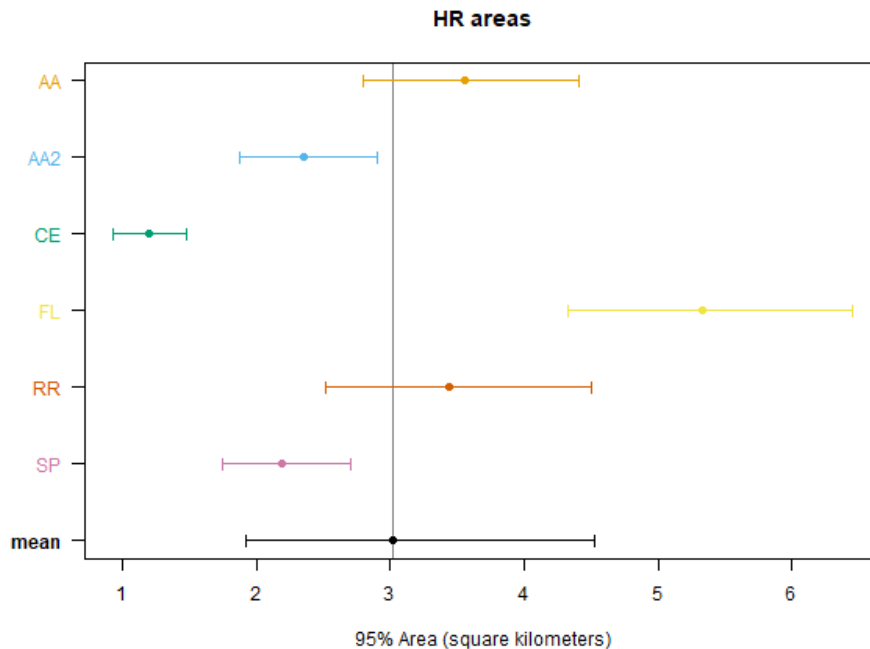


Figure 7: Plot showing the comparison of home range area and confidence intervals for all six complete segments. The colors correspond to the same colors in the home range plots. The mean area across the six complete segments is shown in black on the bottom

References

- Burt, William Henry. 1943. "Territoriality and Home Range Concepts as Applied to Mammals." *Journal of Mammalogy* 24 (3): 346–52. <https://doi.org/10.2307/1374834>.
- Calabrese, Justin M., Chris H. Fleming, and Eliezer Gurarie. 2016. "Ctmm: An r Package for Analyzing Animal Relocation Data as a Continuous-Time Stochastic Process." *Methods in Ecology and Evolution* 7 (9): 1124–32. <https://doi.org/https://doi.org/10.1111/2041-210X.12559>.
- Calabrese, Justin M., Christen H. Fleming, Michael J. Noonan, and Xianghui Dong. 2021. "Ctmmweb: A Graphical User Interface for Autocorrelation-Informed Home Range Estimation." *Wildlife Society Bulletin* 45 (1): 162–69. <https://doi.org/10.1002/wsb.1154>.
- Diggle, Peter, and Paulo J. Ribeiro. 2007. *Model-Based Geostatistics*. Springer Series in Statistics. New York, NY: Springer.
- Fleming, C. H., D. Sheldon, W. F. Fagan, P. Leimgruber, T. Mueller, D. Nandintsetseg, M. J. Noonan, et al. 2018. "Correcting for Missing and Irregular Data in Home-Range Estimation." *Ecological Applications* 28 (4): 1003–10. <https://doi.org/10.1002/eap.1704>.
- Fleming, Christen H., and Justin M. Calabrese. 2017. "A New Kernel Density Estimator for Accurate Home-Range and Species-Range Area Estimation." *Methods in Ecology and Evolution* 8 (5): 571–79. <https://doi.org/10.1111/2041-210X.12673>.
- Fleming, Christen H., Justin M. Calabrese, Thomas Mueller, Kirk A. Olson, Peter Leimgruber, and William F. Fagan. 2014. "Non-Markovian Maximum Likelihood Estimation of Autocorrelated Movement Processes." *Methods in Ecology and Evolution* 5 (5): 462–72. <https://doi.org/10.1111/2041-210X.12176>.
- Fleming, Christen, Justin Calabrese, Thomas Mueller, Kirk Olson, Peter Leimgruber, and William Fagan. 2014. "From Fine-Scale Foraging to Home Ranges: A Semivariance Approach to Identifying Movement Modes Across Spatiotemporal Scales." *The American Naturalist* 183 (May): E154–67. <https://doi.org/10.1086/675504>.
- R Core Team. 2022. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R

Foundation for Statistical Computing. <https://www.R-project.org/>.
Silva, Inês, Christen H. Fleming, Michael J. Noonan, Jesse Alston, Cody Foltz, William Fagan, and Justin M. Calabrese. 2021. “Autocorrelation-Informed Home Range Estimation: A Review and Practical Guide.” Preprint. EcoEvoRxiv. <https://doi.org/10.32942/osf.io/23wq7>.