Black Lion Tamarin Analysis

Ronald Bialozyt

29/08/2019 15:52:27

**Table of Contents**

[Introduction 1](#__RefHeading___Toc9804_2074663610)

[Research Question 1](#__RefHeading___Toc9806_2074663610)

[Methods 1](#__RefHeading___Toc9808_2074663610)

[Read tree data 1](#__RefHeading___Toc9810_2074663610)

[Simulation data 3](#__RefHeading___Toc9812_2074663610)

[Daily path length 3](#__RefHeading___Toc9814_2074663610)

[Homerange 5](#__RefHeading___Toc9816_2074663610)

[Running the simulation 8](#__RefHeading___Toc9818_2074663610)

[Analysing the simulation 8](#__RefHeading___Toc9820_2074663610)

[Observational data 8](#__RefHeading___Toc9822_2074663610)

[Including Plots 8](#__RefHeading___Toc9824_2074663610)

# Introduction

Here we analyse the original and the simulation data from the black lion tamarin.

## Research Question

We would like to find out if the simulation can produce the same pattern.

# Methods

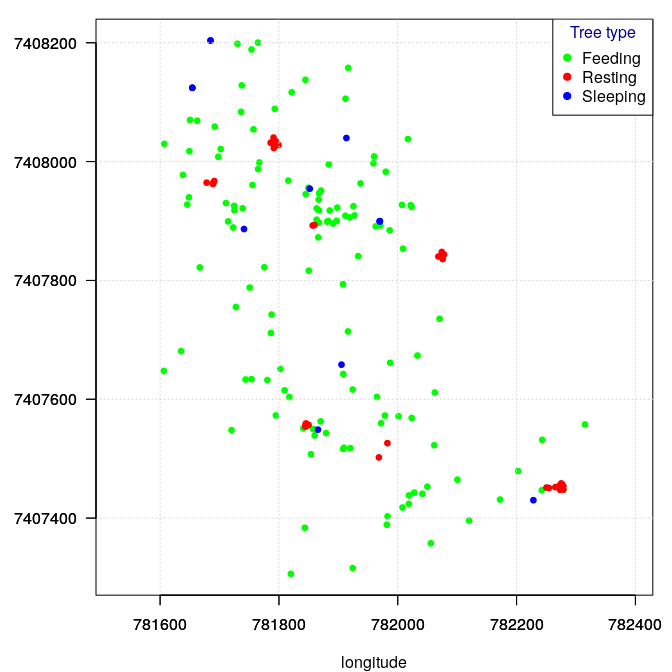
## Read tree data

tree\_file <- "tamarin-trees2.txt"  
trees <- read.table(tree\_file, sep="\t", header=FALSE)  
names(trees) <- c("name", "geo\_x", "geo\_y", "type", "species")  
num.tree.type <- table(trees$type)  
names(num.tree.type) <- c("Feeding","Resting","Sleeping")  
kable(  
 t(num.tree.type), caption = "The amount of trees for each type"  
)

The amount of trees for each type

|  |  |  |
| --- | --- | --- |
| Feeding | Resting | Sleeping |
| 126 | 38 | 13 |

par(mar=c(4,5,1,1))  
par(las=1)  
par(xpd=FALSE)  
## FEEDING RESTING SLEEPING  
cols <- c("green", "red", "blue")  
pchs <- c(19,19,19)  
plot(trees$geo\_x, trees$geo\_y, col=cols[trees$type], asp=1,  
 pch=pchs[trees$type], cex=0.8,  
 xlab="longitude", ylab="" #"latitude"  
 )  
grid(); axis(1); axis(2)  
legend("topright", #703710, 9517990,  
 legend=c("Feeding", "Resting", "Sleeping"),  
 pch=pchs, col=cols, pt.cex=c(1,1,1),  
 bg="white",  
 title="Tree type", title.col="darkblue")



These are the locations of all trees.

We start first with our simulation data from NetLogo and will analyse the observational data afterwards.

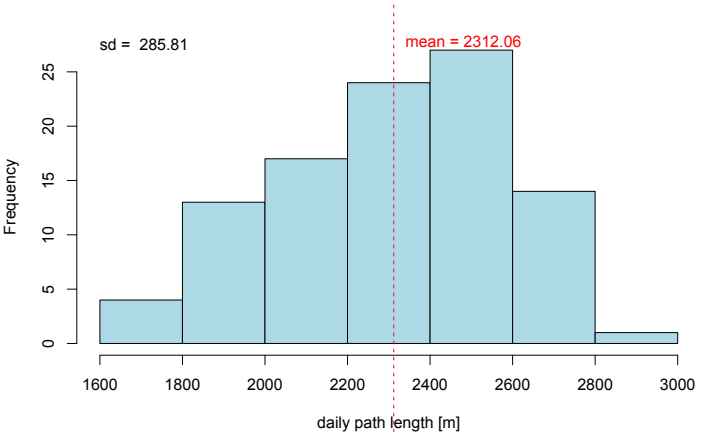
## Simulation data

### Daily path length

f.names <- list.files("./", pattern="monkey\_locations-day\_")  
n.files <- length(f.names)

We did simulate 100 days.

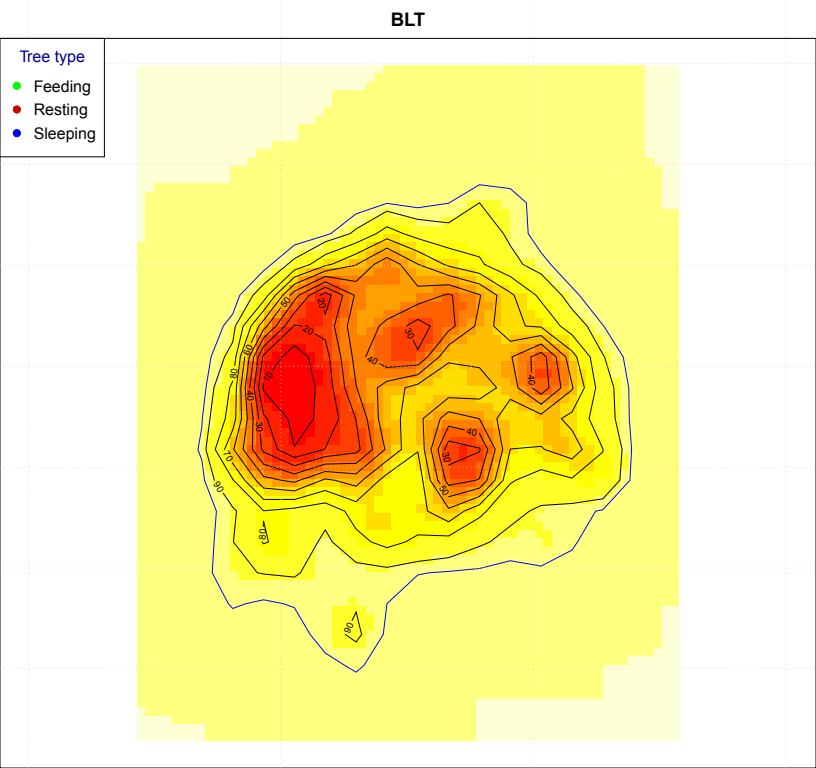
agent\_list <- numeric()  
dist\_list <- numeric()  
all.data <- numeric()  
for(filename in f.names) {  
 ad <- read.table(filename, header=FALSE)  
 names(ad) <- c("time", "day", "timestep",  
 "x\_out", "y\_out", "energy", "target tree",  
 "current tree", "status", "action")  
 all.data <- rbind(all.data, ad)  
 ## == should be now the geographical coordinates  
 ## ad$x\_out <- ad$x\_out \* 25  
 ## ad$y\_out <- ad$y\_out \* 25  
 dist <- 0  
 for(i in 2:nrow(ad)) {  
 dist <- dist + sqrt((ad$x\_out[i] - ad$x\_out[i-1])^2 +  
 (ad$y\_out[i] - ad$y\_out[i-1])^2)  
 }  
 dist\_list <- c(dist\_list, dist)  
 agent\_list <- rbind(agent\_list, ad)  
}  
avg\_dist <- mean(dist\_list)  
sd\_dist <- sd(dist\_list)  
  
par(xpd=TRUE)  
par(oma=c(0,0,0,0))  
par(mar=c(4,4,2,1))  
sim\_hist <- hist(x=dist\_list, ## breaks=seq(500,4000,by=200),  
 col="lightblue",  
 xlab="daily path length [m]", main=""  
 )  
abline(v=avg\_dist, col="red", lty=2)  
text(x=min(sim\_hist$breaks), y=max(sim\_hist$counts)  
 , sprintf("sd = %7.2f", sd\_dist)  
 , adj = c(0, 0))  
text(x=avg\_dist, y=max(sim\_hist$counts)  
 , sprintf("mean = %7.2f", avg\_dist)  
 , col = "red"  
 , adj = c(-0.1, -0.3))



Histogram of the daily path length over 100 simulated days.

### Homerange

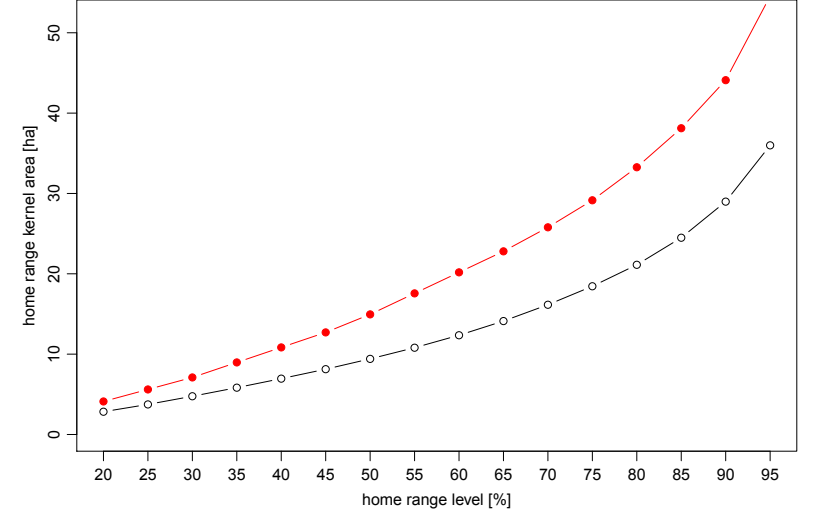
require("adehabitatHR", quietly = TRUE, warn.conflicts = FALSE)  
agent\_data <- agent\_list[,c("x\_out","y\_out")]  
agent\_data$Name <- "BLT"  
coordinates(agent\_data) <- c("x\_out","y\_out")  
  
  
## kud <- kernelUD(agent\_data, h="href")  
kud <- kernelUD(agent\_data, h=38.7) ## taken from Sebastian (and Paper)  
# image(kernelUD(agent\_data, grid=150, extent=0.01, h="href"))  
  
vud <- getvolumeUD(kud)  
  
## image(kernelUD(agent\_data, grid=150, extent=0.01, h="href"))  
par(xpd=TRUE)  
par(oma=c(0,0,0,0))  
image(getvolumeUD(kernelUD(agent\_data, grid=80, extent=0.05, h="href")))  
xyzv <- as.image.SpatialGridDataFrame(vud[[1]])  
contour(xyzv, add=TRUE)  
  
homerange <- getverticeshr(kud)  
# plot(homerange, col=2)  
  
plot(getverticeshr(kud, 95), add=TRUE, lwd=1, border="blue")  
##plot(getverticeshr(kud, 80), add=TRUE, lwd=1, border="red")  
##plot(getverticeshr(kud, 65), add=TRUE, lwd=1, border="green")  
## FEEDING RESTING SLEEPING  
cols <- c("green", "red3", "blue")  
pchs <- c(19,19,19)  
points(trees$geo\_x, trees$geo\_y, col=cols[trees$type], asp=1,  
 pch=pchs[trees$type], cex=0.8,  
 xlab="longitude", ylab="" #"latitude"  
 )  
grid(); ## axis(1); axis(2)  
legend("topleft",  
 ## 703500, 9518300,  
 legend=c("Feeding", "Resting", "Sleeping"), ## 1,2,3  
 pch=pchs, col=cols, pt.cex=c(1,1,1),  
 bg="white",  
 title="Tree type", title.col="darkblue")



The image of the homerange of tamarin group ‘BLT’ from 100 simulation days. The 95% kernel is outlined ‘blue’.

obs.kernel.file <- "observed\_kernelArea\_Field136\_Katrin.txt"  
obs.kernel <- read.table(obs.kernel.file)  
names(obs.kernel) <- "Obs. Kernel"  
sim.kernel <- kernel.area(kud, percent=seq(20, 95, by=5))  
names(sim.kernel) <- "Sim. Kernel"  
tt <- cbind(obs.kernel, sim.kernel)  
tt$perc <- tt[,2] / tt[,1]  
names(tt)[3] <- "Factor"

par(mgp=c(2,0.7,0)) # set the default lines for c(xlab, ylab, main)  
par(oma=c(0,0,0,0))  
par(mar=c(4,4,0,1))  
plot(tt[,1], type="b", ylim=c(0, 52)  
 , xaxt = 'n'   
 , xlab="home range level [%]"  
 , ylab="home range kernel area [ha]")  
axis(1, at = c(1:16), labels = seq(20, 95, by=5))  
points(tt[,2], type="b", pch=19, col="red")



Home range sizes estimated from simulated (red, filled circles) and observed (black, open circles) locations of a group over several days.

kable(tt, digits = c(0,2,2,2), caption = "The homerange sizes of the different kernel percentages. The first column are the values for the observed kernel by Kathrin Lüttmann (2007) and the second one are the values for the simulated kerenl.", label = "kernel\_table")

The homerange sizes of the different kernel percentages. The first column are the values for the observed kernel by Kathrin Lüttmann (2007) and the second one are the values for the simulated kerenl.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Obs. Kernel | Sim. Kernel | Factor |
| 20 | 3 | 4.11 | 1.45 |
| 25 | 4 | 5.61 | 1.50 |
| 30 | 5 | 7.10 | 1.49 |
| 35 | 6 | 8.97 | 1.54 |
| 40 | 7 | 10.84 | 1.56 |
| 45 | 8 | 12.71 | 1.56 |
| 50 | 9 | 14.95 | 1.59 |
| 55 | 11 | 17.57 | 1.63 |
| 60 | 12 | 20.18 | 1.63 |
| 65 | 14 | 22.80 | 1.61 |
| 70 | 16 | 25.79 | 1.60 |
| 75 | 18 | 29.15 | 1.58 |
| 80 | 21 | 33.26 | 1.57 |
| 85 | 24 | 38.12 | 1.56 |
| 90 | 29 | 44.10 | 1.52 |
| 95 | 36 | 54.57 | 1.52 |

### Running the simulation

At the moment per hand and later on from this script.

### Analysing the simulation

We read the simulation data from our files produced by NetLogo.

## Observational data

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.