

Beirne, C., Nuñez, C. L., Baldino, M., Kim, S., Knorr, J., Minich, T., Jin, L., Xiao, S., Mbamy, W., Obiang, G. N., Masseloux, J., Nkoghe, T., Obiang Ebanega, M., Rundel, C., Wright, J. P. and Poulsen, J. R. 2019. Estimation of gut passage time of wild, free roaming forest elephants. – Wildlife Biology 2019: wlb.00543

Appendix 1

GPT trials on captive elephants at the North Carolina Zoo

Previously, Kinahan et al. (2007) fed iButtons embedded in apples to captive African savanna elephants to estimate GPT. We performed a similar baiting trial with three captive savanna elephants at the North Carolina Zoo, Asheboro, NC to test the effectiveness of our encapsulation methods. Encapsulated iButtons were embedded within the seeds of mango fruits and within the flesh of bananas and apples. Fruit cavities were filled with peanut butter to facilitate successful ingestion (Fig. A1). Three encapsulated iButtons were fed to each of three captive adult elephants (two females and one male) during one of their regular feeding times (approximately 9 a.m. on 24 May 2018) within the elephants' holding pens (Fig. A1).



Figure A1. (a) The iButtons (Thermochron DS1921H-F5, Maxim Integrated) were incased within protective epoxy (Loctite Hysol) and labeled to help determine the identity of the consumer (left). (b) We embedded the iButtons within fruits, such as apples, mangoes and bananas (middle). (c) Zookeepers from the North Carolina Zoo fed the baited fruits to individual elephants within their holding pens during one of the regular feeding times.

After the feeding event, all elephant dung located within the elephant enclosures was collected twice daily and searched for the iButtons using a metal detector. Four of the nine iButtons were recovered within five days of the feeding event, three of which were functioning. Using the temperature profiles from the iButtons, GPTs were estimated to be 28.6 and 157.6 h for Tonga and 41.3 h for Batir (Fig. A2).

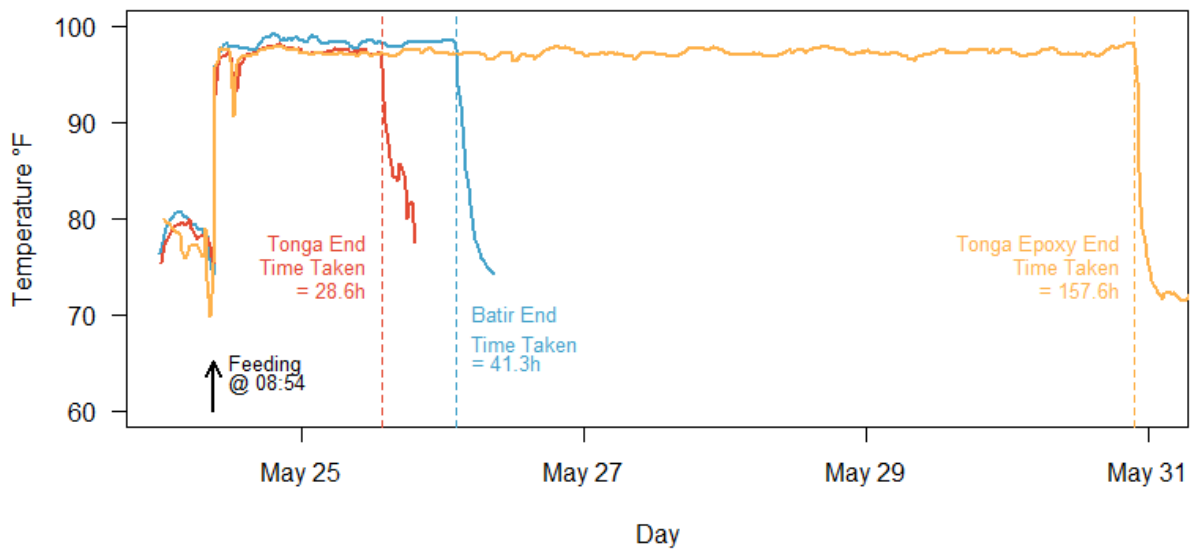


Figure A2. Temperature profiles from the three iButtons retrieved from captive savanna elephants (Tonga and Batir). All iButtons were consumed at 8:54 a.m. on 24 May 2018.

Appendix 2

R code

The code below outlines the moving window analysis used to calculate GPT. It is a nested loop. The outer loop goes through all of the rows of the baiting data frame. If it is the first dung in a follow, the loop uses the start time of the follow as the start time. For all other dungs, the loop uses one day before the end time of the last iteration as the start time. The inner loop goes through all of the rows of the dung data frame when the follow is that of the baiting ID. Together they find the portion of the tracking data such that the tracking start time is greater than or equal to the baiting start time and less than the previous window's ending time, with a six-hour window. The nearest neighbor between the portion of track determined by the moving window and the dung pile is calculated. The latitude, longitude, and time are then added to the dung data frame. Finally, the gut passage time, which is the difference between the dung track time and the bait pile consumption time, is added to the data frame.

```
library(RANN) # needed for nn2 function, which finds the nearest neighbor
library(lubridate) # needed for strptime function,
#which converts from character string to POSIXlt class objects

baiting <- read.csv("Baiting.Locations.csv", header=T)
dung <- read.csv("Dung.Locations.csv", header=T)
track <- read.csv("Elephant.track.csv", header=T)

str(baiting)

baiting$begin <- strptime(baiting$begin, '%Y-%m-%d %H:%M:%S')
baiting$end <- strptime(baiting$end, '%Y-%m-%d %H:%M:%S')
track$time <- strptime(track$time, '%Y-%m-%d %H:%M:%S')
dung$time <- strptime(dung$time, '%Y-%m-%d %H:%M:%S')

dung <- dung[order(dung$follow, dung$time),]
dung.loc <- dung # duplicates dung, one as nearest neighbor recepticle, one as data source
dung$X.track <- NA
dung$Y.track <- NA
dung$track.time <- NA
```

```

# the following loop iterates through each follow

for(j in 1:nrow(baiting)){
  start.baiting <- baiting$end[j]
  eloc.at.dung <- list()

  # if it is the first dung in a follow, use the start time
  # if not use the previous dung encounter (-six hours)
  for(i in 1:nrow(dung[dung$follow==baiting$ID[j],])){

    if(i < 2){ # on the first loop, use the baiting location data to set the starting line
      start <- start.baiting
    }else{
      start <- strptime(dung[dung$follow==baiting$ID[j],]$track.time[i-1],
                        "%Y-%m-%d %H:%M:%S")-(60*60*24) # six hour window
    }

    tmp <- dung.loc[dung.loc$follow==baiting$ID[j],]$time[i]
    #take the first dung find time for that follow

    tmp.trk <- tail(track[track$time < tmp,], n = 1)

    eloc.at.dung[[i]] <- tmp.trk$time

    x <- track[track$time >= start & track$time <= eloc.at.dung[[i],]

    tmp2 <- nn2(x[,c('X', 'Y')], dung.loc[dung.loc$follow==baiting$ID[j],]
               [i ,c('X', 'Y')], k = 1)

    dung[dung.loc$follow==baiting$ID[j],][i, 'X.track'] <- x[tmp2$nn.idx, "X"]
    dung[dung.loc$follow==baiting$ID[j],][i, 'Y.track'] <- x[tmp2$nn.idx, "Y"]
    dung[dung.loc$follow==baiting$ID[j],][i, 'track.time'] <- as.character(x[tmp2$nn.idx,
                                                                              "time"])

  }
}

dung$GPT <- NA # column for GPT calculations

# GPT for first follow

```

```

dung$GPT[dung$follow=="F1"] <- round(difftime(dung$track.time[dung$follow=="F1"],
                                             baiting$end[baiting$ID=="F1"],
                                             units="hours"),2)

# the elephant was lost during the second follow
# therefore this follow does not have enough data to calculate GPT

# GPT for third follow

dung$GPT[dung$follow=="F3"] <- round(difftime(dung$track.time[dung$follow=="F3"],
                                             baiting$end[baiting$ID=="F3"],
                                             units="hours"),2)

# GPT for fourth follow

dung$GPT[dung$follow=="F4"] <- round(difftime(dung$track.time[dung$follow=="F4"],
                                             baiting$end[baiting$ID=="F4"],
                                             units="hours"),2)

# Check Designations
#plotting map with distances
par(mfrow= c(1,2))
cols <- rainbow(nrow(dung))
plot(dung$X, dung$Y, col="white", main = "dung to location attribution", xlab = "X", ylab =
"Y",las=1)
lines(track$X, track$Y, type = "l", col = "grey")

library(geosphere)
library(raster)
for(i in 1: nrow(dung)){
  points(dung$X[i], dung$Y[i], pch = 16, col = cols[i])
  points(dung$X.track[i], dung$Y.track[i], pch = 16, col = cols[i])
  segments(dung[i,"X"],dung[i,"Y"], dung[i,"X.track"],dung[i,"Y.track"], col = cols[i])
  dung$matchDist[i] <- pointDistance(c(dung[i,"X"],dung[i,"Y"]),
c(dung[i,"X.track"],dung[i,"Y.track"]), lonlat=F)
}

#density plot of distances
recept <- matrix(NA, nrow = nrow(dung), ncol = 2)
recept[,1] <- 1:nrow(recept)

```

```
for(i in 1:nrow(recept)){  
  # Works for latLong  
  #recept[i,2] <- distm(c(dung[i,"X"],dung[i,"Y"]), c(dung[i,"X.track"],dung[i,"Y.track"]), fun =  
distGeo)  
  # For UTM  
  recept[i,2] <- pointDistance(c(dung[i,"X"],dung[i,"Y"]), c(dung[i,"X.track"],dung[i,"Y.track"]),  
lonlat=F)  
}  
plot(density(recept[,2]), main = "distance between dung and elephant", xlab = "metres", las=1)  
abline(v=median(recept[,2]), col = "Red")  
text(x = median(recept[,2]), y = .0002, labels = round(median(recept[,2])))
```

Appendix 3

Frequency table of gut passage time (GPT) in hours from dung piles with plastic pellets

The raw data from Figure 4.

GPT interval (h)	Frequency
16.5 – 17.5	3
17.5 – 22.5	8
22.5 – 27.5	13
27.5 – 32.5	19
32.5 – 37.5	15
37.5 – 42.5	10
42.5 – 47.5	16
47.5 – 52.5	7
52.5 – 57.5	12
57.5 – 62.5	5
62.5 – 67.5	1
67.5 – 72.5	3
72.5 – 77.5	1
77.5 – 82.5	2