

Thin Sampling Regimes

Odd Jacobson

2022-12-13

This document shows how I thinned 6 complete segments to create 60 different sampling regimes.

Batch home range calculations at the end.

Load packages

```
library(tidyverse)
library(ctmm)
library(sf)
library(ctmmweb)
library(lubridate)
library(sp)
library(patchwork)
library(RColorBrewer)
```

```
# read in data frame
df <- read.csv("../Data/CH1_GPS_data.csv", row.names = NULL)
```

Thin data sets

Sampling regimes are filtered from the complete segments and converted into telemetry objects to work with the ctmm package.

To filter the data, we use the *lubridate* package to identify time frames to be kept in the different sampling regimes. For concentrated sampling regimes, we filter specific months, weeks, or days to create sampling regimes with 30, 20, 10, 6, and 3 days respectively. We tried to put these sampling regimes at different points in time to minimize overlap between them. Spread sampling regimes were generated by sampling randomly from the complete segments using the *sample* function. We also made these sampling regimes to have 30, 20, 10, 6, and 3 days respectively.

Note that there are two different ways in which we sampled randomly to create the spread sampling regimes. This is because we started off with around 18 sampling regimes and later increased the analysis to 60 sampling regimes to boost our sample size. When we added more sampling regimes, we discovered a better way to generate the spread sampling regimes, but didn't want to change the code in the old ones and change the results.

In the beginning, spread sampling regimes were made by sampling randomly from the total sampling window of the complete segments (i.e. first day to last day). Using this method we had to “over-guess” the amount of days to be selected because there are some missing days in the complete segments, and some randomly selected days did not actually exist in the data. In other words, we would ask for more days to be selected than we wanted in the sampling regime because some days were not in the data. We would repeat the random generation until the amount of days we wanted in the sampling regimes were correct. Later, we

discovered there was an easier way to do this by creating an object with the available days in the complete segment and then randomly sampling from those days. We did not change the old sampling regimes because we did not want to change the results. We used `set.seed` so that the random sampling could be reproduced.

Sampling regimes have what is called an individual.local.identifier which is the unique identifier that ctmm will use to fit a movement model to. The first letter for the the individual.local.identifiers correspond to whether the regimes is concentrated (C) or spread (S), followed by how many days are in the regimes (e.g. C30). The individual.local.identifier for the complete segments are labelled as “all”.

We thin the sampling regimes group by group:

AA Group

```
# subset AA group from dataframe
AA <- df %>%
  filter(group == "AA")

## first object is the complete segment
aa.comp <- AA %>%
  mutate(individual.local.identifier = "all",
         date = date(timestamp)) %>% # add date column
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date") # change to telemetry object for ctmm

# the following objects are thinned within the complete segment

## concentrated regimes.....
# 30 days
aa.c30 <- AA %>%
  filter(week(timestamp) %in% c(1:6),
         date(timestamp) < date("2014-02-09")) %>%
  mutate(individual.local.identifier = "C30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
aa.c20 <- AA %>%
  filter(week(timestamp) %in% c(5:8)) %>%
  mutate(individual.local.identifier = "C20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
aa.c10 <- AA %>%
  filter(week(timestamp) %in% c(2:3),
         date(timestamp) < date("2014-01-18")) %>%
  mutate(individual.local.identifier = "C10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
aa.c6 <- AA %>%
```

```

filter(date(timestamp) > date("2014-02-15")) %>%
mutate(individual.local.identifier = "C6") %>%
as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
aa.c3 <- AA %>%
  filter(week(timestamp) == 5,
         date(timestamp) < date("2014-02-04")) %>%
  mutate(individual.local.identifier = "C3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

## spread sampling regimes.....  

# get available days from complete segment - the following spread sampling regimes will
↪ be randomly sampled from these days
aa_days <- unique(date(aa.comp$timestamp))

# 30 days
set.seed(84) # IMPORTANT: use set.seed for reproducibility in spread samples!
aa.s30 <- AA %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(aa_days, size = 30))) %>%
  mutate(individual.local.identifier = "S30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
set.seed(89)
aa.s20 <- AA %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2014-01-01'), as.Date('2014-02-23'), by="day"),
↪ 25))) %>%
  mutate(individual.local.identifier = "S20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
set.seed(683)
aa.s10 <- AA %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2014-01-01'), as.Date('2014-02-23'), by="day"),
↪ 14))) %>%
  mutate(individual.local.identifier = "S10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
set.seed(510)
aa.s6 <- AA %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2014-01-01'), as.Date('2014-02-23'), by="day"),
↪ 8))) %>%

```

```

  mutate(individual.local.identifier = "S6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
set.seed(43375)
aa.s3 <- AA %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(aa_days, size = 3))) %>%
  mutate(individual.local.identifier = "S3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# combine into list
AA_ctmm <-
  ↵ list(aa.comp,aa.c30,aa.c20,aa.c10,aa.c6,aa.c3,aa.s30,aa.s20,aa.s10,aa.s6,aa.s3)

# rename list elements
names(AA_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
  ↵ "S3")

```

RR Group

```

# subset RR group from dataframe
RR <- df %>%
  filter(group == "RR")

# complete segment
rr.comp <- RR %>% # 39 unique days
  mutate(individual.local.identifier = "all") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

## clumped regimes..... .
# 30 days
rr.c30 <- RR %>%
  filter(week(timestamp) %in% c(15:21),
         date(timestamp) > date("2010-04-12")) %>%
  mutate(individual.local.identifier = "C30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
rr.c20 <- RR %>%
  filter(week(timestamp) %in% c(13:16)) %>%
  mutate(individual.local.identifier = "C20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
rr.c10 <- RR %>%

```

```

filter(week(timestamp) %in% c(18:19)) %>%
  mutate(individual.local.identifier = "C10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
rr.c6 <- RR %>%
  filter(week(timestamp) == 16) %>%
  mutate(individual.local.identifier = "C6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
rr.c3 <- RR %>%
  filter(week(timestamp) == 14,
         date(timestamp) < date("2010-04-05")) %>% # 3 unique days
  mutate(individual.local.identifier = "C3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

## Spread sampling regimes ......

# get available days
rr_days <- unique(date(rr.comp$timestamp))

# 30 days
set.seed(233)
rr.s30 <- RR %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(rr_days, 30))) %>%
  mutate(individual.local.identifier = "S30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
set.seed(111)
rr.s20 <- RR %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(rr_days, 20))) %>%
  mutate(individual.local.identifier = "S20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

set.seed(23)
rr.s10 <- RR %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(seq(as.Date('2010-04-01'), as.Date('2010-05-21'), by="day"),
    ↵ 13))) %>%
  mutate(individual.local.identifier = "S10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
    ↵ +ellps=WGS84 +towgs84=0,0,0")

set.seed(900)

```

```

rr.s6 <- RR %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(seq(as.Date('2010-04-01'), as.Date('2010-05-21'), by="day"),
    10))) %>%
  mutate(individual.local.identifier = "S6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  +ellps=WGS84 +towgs84=0,0,0")

set.seed(853)
rr.s3 <- RR %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(rr_days, 4))) %>%
  mutate(individual.local.identifier = "S3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  +ellps=WGS84 +towgs84=0,0,0")

# combine into list
RR_ctmm <-
  list(rr.comp, rr.c30, rr.c20, rr.c10, rr.c6, rr.c3, rr.s30, rr.s20, rr.s10, rr.s6, rr.s3)

# rename list elements
names(RR_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
  "S3")

```

CE Group

```

# subset CE group from dataframe
CE <- df %>%
  filter(group == "CE")

# complete segment
ce.comp <- CE %>%
  mutate(individual.local.identifier = "all") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  +ellps=WGS84 +towgs84=0,0,0")

## Concentrated regimes .....

# 30 days
ce.c30 <- CE %>%
  filter(week(timestamp) %in% c(28:37),
    date(timestamp) > date("2017-07-13")) %>%
  mutate(individual.local.identifier = "C30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  +ellps=WGS84 +towgs84=0,0,0")

# 20 days
ce.c20 <- CE %>%
  filter(month(timestamp) == 7) %>%
  mutate(individual.local.identifier = "C20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  +ellps=WGS84 +towgs84=0,0,0")

```

```

# 10 days
ce.c10 <- CE %>%
  filter(month(timestamp) == 8,
         date(timestamp) < date("2017-08-18")) %>%
  mutate(individual.local.identifier = "C10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
ce.c6 <- CE %>%
  filter(month(timestamp) == 7,
         date(timestamp) < date("2017-07-08")) %>%
  mutate(individual.local.identifier = "C6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
ce.c3 <- CE %>%
  filter(month(timestamp) %in% c(7,8,9),
         week(timestamp) == 37) %>%
  mutate(individual.local.identifier = "C3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

## Spread sample regimes .....

#get available dates
ce_dates <- unique(date(ce.comp$timestamp))

# 30 days
set.seed(212)
ce.s30 <- CE %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(ce_dates, 30))) %>%
  mutate(individual.local.identifier = "S30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
set.seed(29)
ce.s20 <- CE %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2017-07-01'), as.Date('2017-09-13'), by="day"),
  ↵ 35))) %>%
  mutate(individual.local.identifier = "S20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
set.seed(236)
ce.s10 <- CE %>%
  mutate(date = date(timestamp)) %>%

```

```

filter(date %in% c(sample(seq(as.Date('2017-07-01'), as.Date('2017-09-13'), by="day"),
  ↵  20))) %>%
mutate(individual.local.identifier = "S10") %>%
as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵  +ellps=WGS84 +towgs84=0,0,0")
# 6 days
set.seed(56)
ce.s6 <- CE %>%
  mutate(date = date(timestamp)) %>%
filter(date %in% c(sample(seq(as.Date('2017-07-01'), as.Date('2017-09-13'), by="day"),
  ↵  9))) %>%
  mutate(individual.local.identifier = "S6") %>%
as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵  +ellps=WGS84 +towgs84=0,0,0")

# 3 days
set.seed(8976)
ce.s3 <- CE %>%
  mutate(date = date(timestamp)) %>%
filter(date %in% c(sample(ce_dates, 3))) %>%
  mutate(individual.local.identifier = "S3") %>%
as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵  +ellps=WGS84 +towgs84=0,0,0")

# combine into list
CE_ctmm <-
  ↵  list(ce.comp,ce.c30,ce.c20,ce.c10,ce.c6,ce.c3,ce.s30,ce.s20,ce.s10,ce.s6,ce.s3)

# rename list elements
names(CE_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
  ↵  "S3")

```

AA2 group (AA from a different time period)

```

# subset AA2 from dataframe
AA2 <- df %>%
  filter(group == "AA2")

# complete segment
aa2.comp <- AA2 %>%
  mutate(individual.local.identifier = "all") %>%
as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵  +ellps=WGS84 +towgs84=0,0,0")

## concentrated regimes ......

# 30 days
aa2.c30 <- AA2 %>%
  filter(week(timestamp) %in% c(44:50),
        date(timestamp) < date("2013-12-11")) %>%
  mutate(individual.local.identifier = "C30") %>%

```

```

as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
aa2.c20 <- AA2 %>%
  filter(week(timestamp) %in% c(48,49,50,51),
         date(timestamp) > date("2013-12-02")) %>%
  mutate(individual.local.identifier = "C20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
aa2.c10 <- AA2 %>%
  filter(week(timestamp) %in% c(46,47)) %>%
  mutate(individual.local.identifier = "C10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
aa2.c6 <- AA2 %>%
  filter(week(timestamp) == 45,
         date(timestamp) < date("2013-11-11")) %>%
  mutate(individual.local.identifier = "C6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
aa2.c3 <- AA2 %>%
  filter(week(timestamp) == 47,
         date(timestamp) < date("2013-11-22")) %>%
  mutate(individual.local.identifier = "C3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

## spread sampling regimes ......

# get available dates
aa2_days <- unique(date(aa2.comp$timestamp))

# 30 days
set.seed(815)
aa2.s30 <- AA2 %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(aa2_days, 30))) %>% # 30 days
  mutate(individual.local.identifier = "S30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
set.seed(240)
aa2.s20 <- AA2 %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(seq(as.Date('2013-11-01'), as.Date('2013-12-22'), by="day"),
↪ 25))) %>% # 20 days

```

```

  mutate(individual.local.identifier = "S20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
set.seed(3)
aa2.s10 <- AA2 %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2013-11-01'), as.Date('2013-12-22'), by="day"),
  ↵ 13))) %>% # 11 days
  mutate(individual.local.identifier = "S10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
set.seed(109)
aa2.s6 <- AA2 %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(as.Date('2013-11-01'), as.Date('2013-12-22'), by="day"),
  ↵ 8))) %>% # 7 days
  mutate(individual.local.identifier = "S6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
set.seed(309)
aa2.s3 <- AA2 %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(aa2_days, 3))) %>% # 30 days
  mutate(individual.local.identifier = "S3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# combine into list
AA2_ctmm <-
  ↵ list(aa2.comp,aa2.c30,aa2.c20,aa2.c10,aa2.c6,aa2.c3,aa2.s30,aa2.s20,aa2.s10,aa2.s6,aa2.s3)

# rename list elements
names(AA2_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
  ↵ "S3")

```

SP Group

```

# subset SP group from dataframe
SP <- df %>%
  filter(group == "SP")

# complete segment
sp.comp <- SP %>% # 48 days
  mutate(individual.local.identifier = "all",
         date = date(timestamp)) %>%

```

```

as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

## concentrated regimes..... .

# 30 days
sp.c30 <- SP %>%
  filter(week(timestamp) %in% c(37:46),
         date(timestamp) > date("2010-09-13")) %>%
  mutate(individual.local.identifier = "C30",
         date = date(timestamp)) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 20 days
sp.c20 <- SP %>%
  filter(week(timestamp) %in% c(45:50),
         date(timestamp) > date("2010-11-05")) %>%
  mutate(individual.local.identifier = "C20",
         date = date(timestamp)) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 10 days
sp.c10 <- SP %>%
  filter( week(timestamp) %in% c(40,41,42,43),
          date(timestamp) > date("2010-10-09")) %>%
  mutate(individual.local.identifier = "C10",
         date = date(timestamp)) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 6 days
sp.c6 <- SP %>%
  filter(date(timestamp) < date("2010-09-23")) %>%
  mutate(individual.local.identifier = "C6",
         date = date(timestamp)) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 3 days
sp.c3 <- SP %>%
  filter(week(timestamp) == 47,
         date(timestamp) > date("2010-11-20")) %>%
  mutate(individual.local.identifier = "C3",
         date = date(timestamp)) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

## spread regimes..... .

# get available dates from complete segment
sp_dates <- unique(date(sp.comp$timestamp))

```

```

# 30 days
set.seed(21)
sp.s30 <- SP %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(sp_dates, size = 30))) %>%
  mutate(individual.local.identifier = "S30") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 20 days
set.seed(2109)
sp.s20 <- SP %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(sp_dates, size = 20))) %>%
  mutate(individual.local.identifier = "S20") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 10 days
set.seed(444)
sp.s10 <- SP %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(sp_dates, size = 10))) %>%
  mutate(individual.local.identifier = "S10") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 6 days
set.seed(852)
sp.s6 <- SP %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(sp_dates, size = 6))) %>%
  mutate(individual.local.identifier = "S6") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# 3 days
set.seed(703)
sp.s3 <- SP %>%
  mutate(date = date(timestamp)) %>%
  filter(date %in% c(sample(sp_dates, size = 3))) %>%
  mutate(individual.local.identifier = "S3") %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0")

# combine into list
SP_ctmm <-
  ↵ list(sp.comp, sp.c30, sp.c20, sp.c10, sp.c6, sp.c3, sp.s30, sp.s20, sp.s10, sp.s6, sp.s3)

# rename list elements
names(SP_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
  ↵ "S3")

```

FL Group

```
# subset FL group from dataframe
FL <- df %>%
  filter (group == "FL")

# complete segment
fl.comp <- FL %>% # 64 unique days
  mutate(individual.local.identifier = "all",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# concentrated regimes.....
# 30 days
fl.c30 <- FL %>%
  filter(week(timestamp) %in% c(45:51)) %>%
  mutate(individual.local.identifier = "C30",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 20 days
fl.c20 <- FL %>%
  filter(week(timestamp) %in% c(1,2,3,4),
         date(timestamp) < date("2014-01-24")) %>%
  mutate(individual.local.identifier = "C20",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 10 days
fl.c10 <- FL %>%
  filter(week(timestamp) %in% c(49,50,51),
         date(timestamp) < date("2013-12-14")) %>%
  mutate(individual.local.identifier = "C10",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 6 days
fl.c6 <- FL %>%
  filter( week(timestamp) %in% c(45,46),
         date(timestamp) < date("2013-11-17")) %>%
  mutate(individual.local.identifier = "C6",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")
```

```

# 3 days
fl.c3 <- FL %>%
  filter(week(timestamp) == 7,
         date(timestamp) < date("2014-02-16")) %>%
  mutate(individual.local.identifier = "C3",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# spread regimes.....  

# get fl available dates
fl_dates <- unique(date(f1.comp$timestamp))

# 30 days
set.seed(389)
fl.s30 <- FL %>%
  filter(date(timestamp) %in% c(sample(f1_dates, size = 30))) %>%
  mutate(individual.local.identifier = "S30",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 20 days
set.seed(710)
fl.s20 <- FL %>%
  filter(date(timestamp) %in% c(sample(f1_dates, size = 20))) %>%
  mutate(individual.local.identifier = "S20",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 10 days
set.seed(656)
fl.s10 <- FL %>%
  filter(date(timestamp) %in% c(sample(f1_dates, size = 10))) %>%
  mutate(individual.local.identifier = "S10",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
  ↵ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 6 days
set.seed(3449)
fl.s6 <- FL %>%
  filter(date(timestamp) %in% c(sample(f1_dates, size = 6))) %>%
  mutate(individual.local.identifier = "S6",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%

```

```

as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# 3 days
set.seed(5993)
fl.s3 <- FL %>%
  filter(date(timestamp) %in% c(sample(fl_dates, size = 3))) %>%
  mutate(individual.local.identifier = "S3",
         date = date(timestamp)) %>%
  arrange(timestamp) %>%
  as.telemetry(projection = "+proj=utm +zone=16 +north +datum=WGS84 +units=m +no_defs
↪ +ellps=WGS84 +towgs84=0,0,0", keep = "date")

# combine into list
FL_ctmm <-
↪ list(f1.comp,f1.c30,f1.c20,f1.c10,f1.c6,f1.c3,f1.s30,f1.s20,f1.s10,f1.s6,f1.s3)

# rename list elements
names(FL_ctmm) <- c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20", "S10", "S6",
↪ "S3")

```

Temporal Plots

Function to get global axes

```

## function to get global axes
add_global_label <- function(pwobj, Xlab = NULL, Ylab = NULL, Xgap = 0.03, Ygap = 0.03,
↪ ...) {
  ylabgrob <- patchwork::plot_spacer()
  if (!is.null(Ylab)) {
    ylabgrob <- ggplot() +
      geom_text(aes(x = .5, y = .5), label = Ylab, angle = 90, ...) +
      theme_void()
  }
  if (!is.null(Xlab)) {
    xlabgrob <- ggplot() +
      geom_text(aes(x = .5, y = .5), label = Xlab, ...) +
      theme_void()
  }
  if (!is.null(Ylab) & is.null(Xlab)) {
    return((ylabgrob + patchworkGrob(pwobj)) +
           patchwork::plot_layout(widths = 100 * c(Ygap, 1 - Ygap)))
  }
  if (is.null(Ylab) & !is.null(Xlab)) {
    return((ylabgrob + pwobj) +
           (xlabgrob) +
           patchwork::plot_layout(heights = 100 * c(1 - Xgap, Xgap),
                                 widths = c(0, 100),
                                 design =
                                   AB
                                   CC
                                   ""))
  }
}

```

```

        ))
    }
    if (!is.null(Ylab) & !is.null(Xlab)) {
      return((ylabgrob + pwobj) +
             (xlabgrob) +
             patchwork:::plot_layout(heights = 100 * c(1 - Xgap, Xgap),
                                     widths = 100 * c(Ygap, 1 - Ygap),
                                     design = "
                                         AB
                                         CC
                                         "
                                     ))
    }
    return(pwobj)
}

```

Customize color palettes

```

# make custom color palettes
greens <- colorRampPalette(brewer.pal(8, "Greens"))(20)[6:20]
oranges <- colorRampPalette(brewer.pal(8, "Oranges"))(20)[6:20]
blues <- colorRampPalette(brewer.pal(8, "Blues"))(20)[6:20]
purples <- colorRampPalette(brewer.pal(8, "Purples"))(20)[6:20]
reds <- rev(colorRampPalette(hcl.colors(8, "Reds 3")))(30)[10:21]) # reverse order so fade
  ↵ is consistent with brewer.pal
browns <- rev(colorRampPalette(hcl.colors(8, "BrwnYl")))(20)[4:16])

```

Make plots

```

p1 <- AA_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ↵ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("AA Group") +
  scale_fill_manual(values = greens) +
  scale_y_continuous(breaks=c(0,25,50)) +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        legend.position = "none",
        axis.text = element_text(size = 8),
        axis.title = element_blank())

p2 <- CE_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ↵ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("CE Group") +

```

```

scale_fill_manual(values = oranges) +
scale_y_continuous(breaks=c(0,25,50)) +
theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
      legend.position = "none",
      axis.text = element_text(size = 8),
      axis.title = element_blank())

p3 <- RR_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ~ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("RR Group") +
  scale_fill_manual(values = blues) +
  scale_y_continuous(breaks=c(0,25,50)) +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        legend.position = "none",
        axis.text = element_text(size = 8),
        axis.title = element_blank())


p4 <- AA2_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ~ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("AA2 Group") +
  scale_fill_manual(values = purples) +
  scale_y_continuous(breaks=c(0,25,50)) +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        legend.position = "none",
        axis.text = element_text(size = 8),
        axis.title = element_blank())


p5 <- SP_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ~ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("SP Group") +
  scale_fill_manual(values = reds) +
  scale_y_continuous(breaks=c(0,25,50)) +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        legend.position = "none",
        axis.text = element_text(size = 8),
        axis.title = element_blank())

```

```

p6 <- FL_ctmm %>%
  collect() %>%
  purrr::pluck("data_dt") %>%
  mutate(id = factor(id, levels = c("all", "C30", "C20", "C10", "C6", "C3", "S30", "S20",
  ↵ "S10", "S6", "S3"))) %>%
  plot_time() +
  theme_bw() +
  ggtitle("FL Group") +
  scale_fill_manual(values = browns) +
  scale_y_continuous(breaks=c(0,25,50)) +
  theme(plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
        legend.position = "none",
        axis.text = element_text(size = 8),
        axis.title = element_blank())

```

Combine and save

```

regime_plots <- ((p1 + p2 + p3 + p4 + p5 + p6) + plot_layout(ncol = 3)) %>%
  add_global_label(Ylab = "Data Count",
                   Xlab = "Time",
                   size = 5)

```

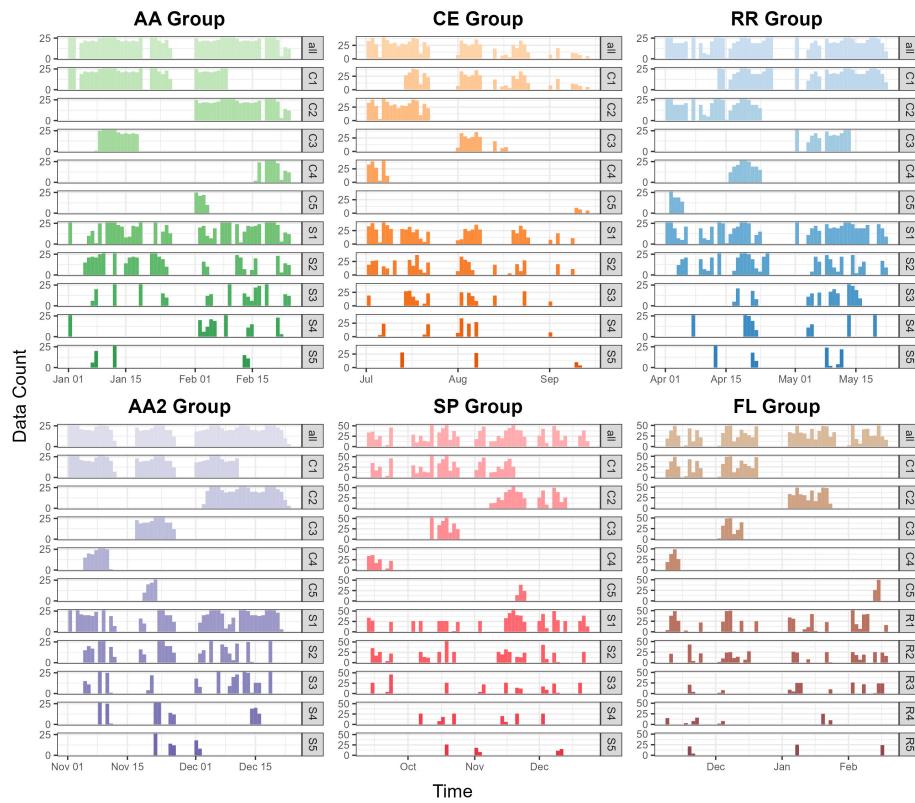


Figure 1: Thinned sampling regimes

Plot full dataset with complete segments boxed – full dataset not available to public so won't be able to reproduce this plot

```

# read in data saved from above formatting
data <- readRDS("../Data/30min_trkpts_formatted") %>% # read in full dataset at 30min
  ↪ sampling rate
  mutate(group = str_sub(individual.local.identifier, 1,2),
         id = str_c(group,"_ ",year,"_ ",season))

# summarize number of observations per date per group
d_sum <- data %>%
  mutate(date = date(timestamp)) %>%
  group_by(date, group) %>%
  summarise(n = length(timestamp))

```

```

# plot
full_data_plot <- d_sum %>%
  ggplot(aes(x = date, y = n/26)) +
  geom_hline(aes(yintercept = 0), color = "black") +
  geom_hline(aes(yintercept = 1), color = "gray70", lty = 3) +
  geom_segment(aes(xend = date, yend = 0), alpha = 0.5) +
  geom_rect(data = subset(d_sum, group == "AA"),
            size = 1.3,
            fill = NA, colour = "#009E73",
            xmin = min(as.Date((AA$timestamp))), xmax = max(as.Date((AA$timestamp))),
            ymin = 0,ymax = 1,
            show.legend = FALSE) +
  geom_label(data = subset(d_sum, group == "AA"),
             aes(x = max(as.Date((AA$timestamp))) + 100,
                  y = 0.5),
             fill = "#009E73",
             label = "AA") +
  geom_rect(data = subset(d_sum, group == "CE"),
            size = 1.3,
            fill = NA, colour = "#E69F00",
            xmin = min(as.Date((CE$timestamp))), xmax = max(as.Date((CE$timestamp))),
            ymin = 0,ymax = 1,
            show.legend = FALSE) +
  geom_label(data = subset(d_sum, group == "CE"),
             aes(x = max(as.Date((CE$timestamp))) + 100,
                  y = 0.5),
             fill = "#E69F00",
             label = "CE") +
  geom_rect(data = subset(d_sum, group == "RR"),
            size = 1.3,
            fill = NA, colour = "#56B4E9",
            xmin = min(as.Date((RR$timestamp))), xmax = max(as.Date((RR$timestamp))),
            ymin = 0,ymax = 1,
            show.legend = FALSE) +
  geom_label(data = subset(d_sum, group == "RR"),
             aes(x = max(as.Date((RR$timestamp))) + 100,
                  y = 0.5),
             fill = "#56B4E9",
             label = "RR") +
  geom_rect(data = subset(d_sum, group == "AA"),
            size = 1.3,
            fill = NA, colour = "#009E73",
            xmin = min(as.Date((AA$timestamp))), xmax = max(as.Date((AA$timestamp))),
            ymin = 0,ymax = 1,
            show.legend = FALSE)

```

```

    fill = NA, colour = "#AA4499",
    xmin = min(as.Date((AA2$timestamp))), xmax = max(as.Date((AA2$timestamp)) -
    ↵ 3),
    ymin = 0,ymax = 1,
    show.legend = FALSE) +
geom_label(data = subset(d_sum, group == "AA"),
           aes(x = max(as.Date((AA2$timestamp))) - 170,
               y = 0.5),
           fill = "#AA4499",
           label = "AA2") +
geom_rect(data = subset(d_sum, group == "FL"),
           size = 1.3,
           fill = NA, colour = "#997700",
           xmin = min(as.Date((FL$timestamp))), xmax = max(as.Date((FL$timestamp))),
           ymin = 0,ymax = 1,
           show.legend = FALSE) +
geom_label(data = subset(d_sum, group == "FL"),
           aes(x = max(as.Date((FL$timestamp))) + 100,
               y = 0.5),
           fill = "#997700",
           label = "FL") +
geom_rect(data = subset(d_sum, group == "SP"),
           size = 1.3,
           fill = NA, colour = "#D55E00",
           xmin = min(as.Date((SP$timestamp))), xmax = max(as.Date((SP$timestamp))),
           ymin = 0,ymax = 1,
           show.legend = FALSE) +
geom_label(data = subset(d_sum, group == "SP"),
           aes(x = max(as.Date((SP$timestamp))) + 100,
               y = 0.5),
           fill = "#D55E00",
           label = "SP") +
theme_bw() +
theme(panel.border = element_blank(),
      panel.grid.major.y = element_blank(),
      panel.grid.minor.y = element_blank(),
      panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank(),
      axis.text.y = element_text(size = 7),
      legend.position = "none",
      plot.title = element_text(hjust = 0.5)) +
scale_y_continuous(limits = c(0,1), breaks = 1, labels = "13h") +
scale_x_date(breaks = scales::date_breaks("1 years"), labels =
    ↵ scales::date_format("%Y")) +
labs(x = "Date", y = "") +
theme(axis.text.y = element_text(size = 12)) +
facet_grid(group ~ ., drop = TRUE)

```

Batch Home Range Estimation

The following code generates the variograms, movement models, and akde home ranges in a single loop per group. These calculation take a long time to run. The results are saved in the Intermediate folder of the

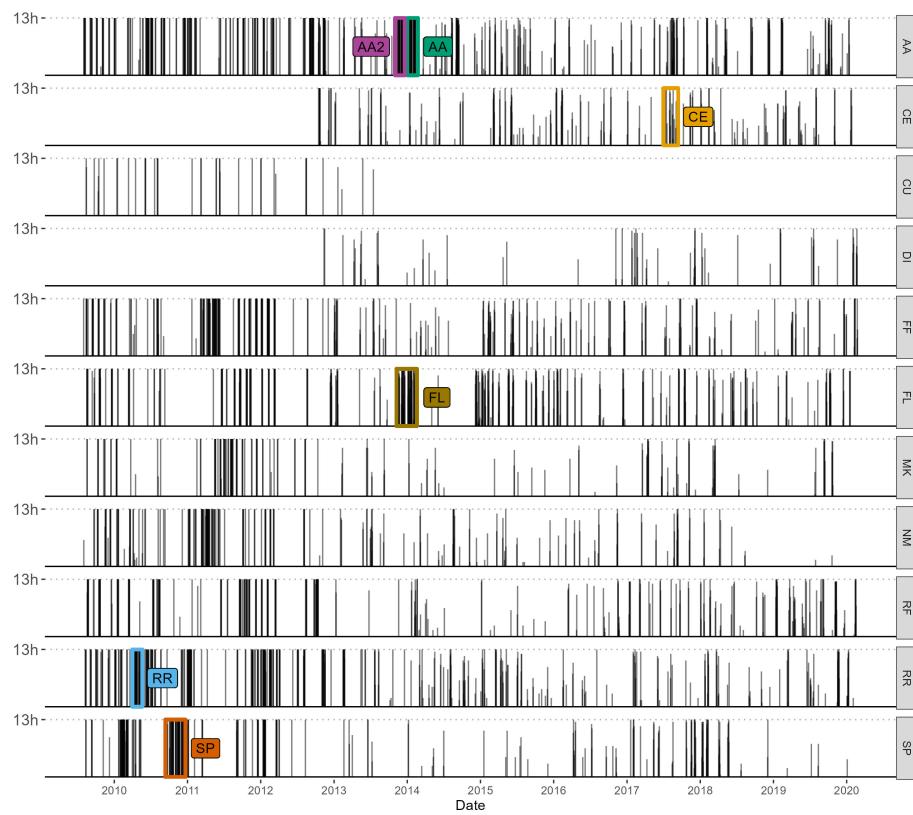


Figure 2: Full dataset with the sampling regimes put in rectangles

github repository if one would like to skip this step. Also, see supplementary file X for detailed walk through of this analysis using an example from one complete segment.

```
# BATCH ANALYSIS - VARIOGRAM, MODEL FITTING, AKDE ##

# AA.....
SVFa <- FITSa <- AKDEa <- list() # create empty lists to be filled

for(i in 1:length(AA_ctmm)){
  SVFa[[i]] <- variogram(AA_ctmm[[i]])
  GUESS <- ctmm.guess(AA_ctmm[[i]],interactive=FALSE, variogram = SVFa[[i]]) #use
  → variogram to get initial parameters
  FITSa[[i]] <- ctmm.select(AA_ctmm[[i]],GUESS,trace=2) # fit movement models
  AKDEa[[i]] <- akde(AA_ctmm[[i]],FITSa[[i]], weights = TRUE,
  → grid=list(dr=10,align.to.origin=TRUE)) # hr estimation
}

# note: weights argument offsets irregular data, and grid aligns the akdes so overlap
→ function is possible

# give names same as data
names(AKDEa) <- names(FITSa) <- names(SVFa) <- names(AA_ctmm)

#RR.....
SVFr <- FITSr <- AKDER <- list()

for(i in 1:length(RR_ctmm)){
  SVFr[[i]] <- variogram(RR_ctmm[[i]])
  GUESS <- ctmm.guess(RR_ctmm[[i]],interactive=FALSE, variogram = SVFr[[i]])
  FITSr[[i]] <- ctmm.select(RR_ctmm[[i]],GUESS,trace=2)
  AKDER[[i]] <- akde(RR_ctmm[[i]],FITSr[[i]], weights = TRUE,
  → grid=list(dr=10,align.to.origin=TRUE))
}

names(AKDER) <- names(FITSr) <- names(SVFr) <- names(RR_ctmm)

# CE.....
SVFc <- FITSc <- AKDEC <- list()

for(i in 1:length(CE_ctmm)){
  SVFc[[i]] <- variogram(CE_ctmm[[i]])
  GUESS <- ctmm.guess(CE_ctmm[[i]],interactive=FALSE, variogram = SVFc[[i]])
  FITSc[[i]] <- ctmm.select(CE_ctmm[[i]],GUESS,trace=2)
  AKDEC[[i]] <- akde(CE_ctmm[[i]],FITSc[[i]], weights = TRUE,
  → grid=list(dr=10,align.to.origin=TRUE))
}

names(AKDEC) <- names(FITSc) <- names(SVFc) <- names(CE_ctmm)

# AA2.....
SVFa2 <- FITSa2 <- AKDEa2 <- list()

for(i in 1:length(AA2_ctmm)){
  SVFa2[[i]] <- variogram(AA2_ctmm[[i]])
```

```

GUESS <- ctmm.guess(AA2_ctmm[[i]],interactive=FALSE, variogram = SVFa2[[i]])
FITSa2[[i]] <- ctmm.select(AA2_ctmm[[i]],GUESS,trace=2)
AKDEa2[[i]] <- akde(AA2_ctmm[[i]],FITSa2[[i]], weights = TRUE,
↪ grid=list(dr=10,align.to.origin=TRUE))
}

# give names same as data
names(AKDEa2) <- names(FITSa2) <- names(SVFa2) <- names(AA2_ctmm)

# SP.....
SVFs <- FITSs <- AKDEs <- list()

for(i in 1:length(SP_ctmm)){
  SVFs[[i]] <- variogram(SP_ctmm[[i]])
  GUESS <- ctmm.guess(SP_ctmm[[i]],interactive=FALSE, variogram = SVFs[[i]])
  FITSs[[i]] <- ctmm.select(SP_ctmm[[i]],GUESS,trace=2)
  AKDEs[[i]] <- akde(SP_ctmm[[i]],FITSs[[i]], weights = TRUE,
↪ grid=list(dr=10,align.to.origin=TRUE))
}

names(AKDEs) <- names(FITSs) <- names(SVFs) <- names(SP_ctmm)

# FL .....
SVFf <- FITSf <- AKDEF <- list()

for(i in 1:length(FL_ctmm)){
  SVFf[[i]] <- variogram(FL_ctmm[[i]])
  GUESS <- ctmm.guess(FL_ctmm[[i]],interactive=FALSE, variogram = SVFf[[i]])
  FITSf[[i]] <- ctmm.select(FL_ctmm[[i]],GUESS,trace=2)
  AKDEF[[i]] <- akde(FL_ctmm[[i]],FITSf[[i]], weights = TRUE,
↪ grid=list(dr=10,align.to.origin=TRUE))
}

names(AKDEF) <- names(FITSf) <- names(SVFf) <- names(FL_ctmm)

```