#You need to install of these libraries in order to run the functions

library(tidyverse);

library(lubridate);

library(ctmm);

library(parallel);

library(foreach);

library(gridExtra)

library(dplyr)

library(stringr)

doParallel::registerDoParallel(4)

#Examples of how to run all the functions are at the bottom of the script

compare\_overlap <- function(before\_model, after\_model){

#Takes in two lists.

#Before\_model takes a list comtaining two fitted models of the individuals before encounter.

#After\_model is the list of fitted models after encounter

#Returns a data.frame that lets you compare values easier

before\_overlap <- overlap(before\_model)

after\_overlap <- overlap(after\_model)

low\_comp <- c(before\_overlap[2], after\_overlap[2])

est\_comp <- c(before\_overlap[6], after\_overlap[6])

high\_comp <- c(before\_overlap[10], after\_overlap[10])

return(data.frame( rows = c("before","after"), low = low\_comp, est = est\_comp, high = high\_comp,

row.names = "rows"))

}

----------------------------------------------

#The following is a script that runs all the AKDE analysis and saves all the data into a folder it creates

#This splits the data EXACTLY before and after the encounter time, it does not factor in "transition times"

#To factor in transitions for on encounter, use run\_akde\_with\_data instead

#You need to input the following

#name1: the name of individual one (make sure it's spelled exactly)

#name2: the name of individual two (make sure it's spelled exaclty)

#study: the study the pair is from

#date\_encounter: the date of encounter as a lubridate datetime (use ymd\_hms("2016-04-23 14:30:00") to create them)

run\_akde\_analysis <- function(name1, name2, study, date\_encounter){

folder\_name <- str\_c(name1, "\_", name2, "\_Homeranges", sep = "")

#Change the folder\_dir string to the folder where you want all your results to be stored

#Do not add any slashes to the end of the string

#You ONLY need to change this variable

folder\_dir <- "C:/workhard/Movement Seminar/MovementSeminar/Code\_Mingxi"

folder\_path <- str\_c(folder\_dir, folder\_name, sep = "\\")

dir.create(folder\_path)

id1 <- movement\_data\_id\_all4 %>% filter(study.id == study, individual.local.identifier ==name1)

id2 <- movement\_data\_id\_all4 %>% filter(study.id == study,individual.local.identifier == name2)

before <- list(filter(id1, timestamp < date\_encounter),

filter(id2, timestamp < date\_encounter)) %>% map(., as.telemetry)

after <- list(filter(id1, timestamp > date\_encounter),

filter(id2, timestamp > date\_encounter)) %>% map(., as.telemetry)

save(before, file = str\_c(folder\_path, "\\", folder\_name, "\_before\_data.rda"))

save(after, file = str\_c(folder\_path, "\\", folder\_name, "\_after\_data.rda"))

#Sets the projections to be the same as before. Change both to after is you want them standardized to after

projection(before) <- median(before)

projection(after) <- median(before)

#Print the number of entries in each split dataset, gives you a sense of scale

print(map\_dbl(before, nrow))

print(map\_dbl(after, nrow))

GUESS <- lapply(Hedley\_Diabla\_before[1:2], function(b) ctmm.guess(b,interactive=FALSE) )

FITS\_Hedley\_Diabla\_before <- lapply(1:2, function(i) ctmm.select(Hedley\_Diabla\_before[[i]],GUESS[[i]]) )

akde\_all <- function(data){

guess <- map(data, ctmm.guess, interactive = FALSE)

fit <- map2(data, guess, ctmm.select)

akde\_final <- map2(data, fit, akde)

return(list(fit = fit, akde = akde\_final))

}

seq\_tele <- list(before = before, after = after)

model\_list <- foreach ( i = seq\_along(seq\_tele), .packages = c("ctmm","lubridate","tidyverse")) %dopar%

{akde\_all(seq\_tele[[i]])}

before\_model <- model\_list[[1]]

after\_model <- model\_list[[2]]

save(before\_model, file = str\_c(folder\_path, "\\", folder\_name, "\_before\_model.rda", sep = ""))

save(after\_model, file = str\_c(folder\_path, "\\", folder\_name, "\_after\_model.rda", sep = ""))

#Overlap analysis table

overlap\_analysis <- compare\_overlap(before\_model$fit, after\_model$fit)

png(filename = str\_c(folder\_path, "\\", folder\_name,"\_overlap\_table.png", sep = ""),

height = 50\*nrow(overlap\_analysis), width = 200\*ncol(overlap\_analysis))

grid.table(overlap\_analysis)

dev.off()

#Save Overlap Table as dataframe

save(overlap\_analysis, file = str\_c(folder\_path, "\\", folder\_name,"\_overlap\_table.rda", sep = ""))

pair\_names <- names(before\_model$akde)

#Make the effective home range table for reporting

eff\_home\_range <- data.frame(x = c(before\_model$akde[[1]]$DOF.area[1],

after\_model$akde[[1]]$DOF.area[1]),

y = c(before\_model$akde[[2]]$DOF.area[1],

after\_model$akde[[2]]$DOF.area[1]))

colnames(eff\_home\_range) <- pair\_names

rownames(eff\_home\_range) <- c("Before","After")

png(filename = str\_c(folder\_path, "\\", folder\_name,"\_effective\_homerange\_table.png", sep = ""),

height = 50\*nrow(eff\_home\_range), width = 200\*ncol(eff\_home\_range))

grid.table(eff\_home\_range)

dev.off()

save(eff\_home\_range, file = str\_c(folder\_path, "\\", folder\_name,"\_effective\_homerange\_table.rda", sep = ""))

extents <- rbind(ctmm::extent(before\_model$fit[[1]]), ctmm::extent(before\_model$fit[[2]]),

ctmm::extent(before\_model$akde[[1]]), ctmm::extent(before\_model$akde[[2]]),

ctmm::extent(after\_model$fit[[1]]), ctmm::extent(after\_model$fit[[2]]),

ctmm::extent(after\_model$akde[[1]]), ctmm::extent(after\_model$akde[[2]]))

x\_range <- c((min(extents$x) - 500 %#% "meters"),(max(extents$x) + 500 %#% "meters"))

y\_range <- c((min(extents$y) - 500 %#% "meters"),(max(extents$y) + 500 %#% "meters"))

sorted\_names <- sort(c(name1, name2), decreasing = F)

#Before encounter home range estimates

png(filename = str\_c(folder\_path, "\\", folder\_name,"\_before.png", sep = ""),

width = 13, height = 6, units = "in", res = 75)

plot(before, col = color(before, by = "individual"), UD = before\_model$akde,

ylim = y\_range, xlim = x\_range, level.UD = 0.95,

main = str\_c(name1, " and ", name2, " before encounter: ",date\_encounter))

legend("bottomleft", legend = sorted\_names, col = c("red","blue"), pch = 1, cex =2.5)

dev.off()

#After encounter home range estimates

png(filename = str\_c(folder\_path, "\\", folder\_name,"\_after.png", sep = ""),

width = 13, height = 6, units = "in", res = 75)

plot(after, col = color(after, by = "individual"), UD = after\_model$akde,

ylim = y\_range, xlim = x\_range,

main = str\_c(name1, " and ", name2, " after encounter: ",date\_encounter))

legend("bottomleft", legend = sorted\_names, col = c("red","blue"), pch = 1, cex = 2.5)

dev.off()

}

---------------------------------------------

#This runs that same AKDE analysis as in the function above, expect you give it the data list to run the analysis on

#This allows you to split and discard some data how you like

#It does not work with pairs with multiple encounter times

#You must input

#before\_list: a list containing the data frames of both individual's before data. It must be names corresponding to the names of the pairs

#after\_list: same as before list except it contains the data after the encounter

#date\_encounter: the date of the encounter, formatted as a lubridate datetime (use ymd\_hms() function)

run\_akde\_with\_data <- function(before\_list, after\_list, date\_encounter){

pair\_names <- names(before\_list)

name1 <- pair\_names[1]

name2 <- pair\_names[2]

folder\_name <- str\_c(name1, "\_", name2, "\_Homeranges", sep = "")

#Change the folder\_dir string to the folder where you want all your results to be stored

#Do not add any slashes to the end of the string

#You ONLY need to change this variable

folder\_dir <- "C:/workhard/Movement Seminar/MovementSeminar/Code\_Mingxi"

folder\_path <- str\_c(folder\_dir, folder\_name, sep = "/")

dir.create(folder\_path)

before <- before\_list %>% map(., as.telemetry)

after <- after\_list %>% map(., as.telemetry)

#Sets the projections to be the same as before. Change both to after is you want them standardized to after

projection(before[1]) <- median(before)

projection(before[2]) <- median(before)

projection(after[1]) <- median(before)

projection(after[2]) <- median(before)

#Print the number of entries in each split dataset, gives you a sense of scale

print(map\_dbl(before, nrow))

print(map\_dbl(after, nrow))

akde\_all <- function(data){

guess <- map(data, ctmm.guess, interactive = FALSE)

fit <- map2(data, guess, ctmm.select)

akde\_final <- map2(data, fit, akde)

return(list(fit = fit, akde = akde\_final))

}

seq\_tele <- list(before = before, after = after)

model\_list <- foreach ( i = seq\_along(seq\_tele), .packages = c("ctmm","lubridate","tidyverse")) %dopar%

{akde\_all(seq\_tele[[i]])}

before\_model <- model\_list[[1]]

after\_model <- model\_list[[2]]

save(before\_model, file = str\_c(folder\_path, "/", folder\_name, "\_before\_model.rda", sep = ""))

save(after\_model, file = str\_c(folder\_path, "/", folder\_name, "\_after\_model.rda", sep = ""))

#Overlap analysis table

overlap\_analysis <- compare\_overlap(before\_model$fit, after\_model$fit)

png(filename = str\_c(folder\_path, "/", folder\_name,"\_overlap\_table.png", sep = ""),

height = 50\*nrow(overlap\_analysis), width = 200\*ncol(overlap\_analysis))

grid.table(overlap\_analysis)

dev.off()

#Save Overlap Table as dataframe

save(overlap\_analysis, file = str\_c(folder\_path, "/", folder\_name,"\_overlap\_table.rda", sep = ""))

pair\_names <- names(before\_model$akde)

#Make the effective home range table for reporting

eff\_home\_range <- data.frame(x = c(before\_model$akde[[1]]$DOF.area[1],

after\_model$akde[[1]]$DOF.area[1]),

y = c(before\_model$akde[[2]]$DOF.area[1],

after\_model$akde[[2]]$DOF.area[1]))

colnames(eff\_home\_range) <- pair\_names

rownames(eff\_home\_range) <- c("Before","After")

png(filename = str\_c(folder\_path, "/", folder\_name,"\_effective\_homerange\_table.png", sep = ""),

height = 50\*nrow(eff\_home\_range), width = 200\*ncol(eff\_home\_range))

grid.table(eff\_home\_range)

dev.off()

save(eff\_home\_range, file = str\_c(folder\_path, "/", folder\_name,"\_effective\_homerange\_table.rda", sep = ""))

extents <- rbind(ctmm::extent(before\_model$fit[[1]]), ctmm::extent(before\_model$fit[[2]]),

ctmm::extent(before\_model$akde[[1]]), ctmm::extent(before\_model$akde[[2]]),

ctmm::extent(after\_model$fit[[1]]), ctmm::extent(after\_model$fit[[2]]),

ctmm::extent(after\_model$akde[[1]]), ctmm::extent(after\_model$akde[[2]]))

x\_range <- c((min(extents$x) - 500 %#% "meters"),(max(extents$x) + 500 %#% "meters"))

y\_range <- c((min(extents$y) - 500 %#% "meters"),(max(extents$y) + 500 %#% "meters"))

sorted\_names <- sort(pair\_names, decreasing = F)

#Before encounter home range estimates

png(filename = str\_c(folder\_path, "/", folder\_name,"\_before.png", sep = ""),

width = 13, height = 6, units = "in", res = 75)

plot(before, col = color(before, by = "individual"), UD = before\_model$akde,

ylim = y\_range, xlim = x\_range, level.UD = 0.95,

main = str\_c(name1, " and ", name2, " before encounter: ",date\_encounter))

legend("bottomleft", legend = sorted\_names, col = c("red","blue"), pch = 1, cex =2.5)

dev.off()

#After encounter home range estimates

png(filename = str\_c(folder\_path, "/", folder\_name,"\_after.png", sep = ""),

width = 13, height = 6, units = "in", res = 75)

plot(after, col = color(after, by = "individual"), UD = after\_model$akde,

ylim = y\_range, xlim = x\_range,

main = str\_c(name1, " and ", name2, " after encounter: ",date\_encounter))

legend("bottomleft", legend = sorted\_names, col = c("red","blue"), pch = 1, cex = 2.5)

dev.off()

}

#Examples-------------------------------

#make sure you have the movement\_data\_id\_all4 dataset loaded in before trying these examples

#run\_akde\_analysis example

#make datetime to use as date\_encounter

#Apethe\_Merimela\_encounter\_date <- ymd\_hms("2016-08-12 03:00:00")

#run\_akde\_analysis("Apethe", "Merimela", "Walton", Apethe\_Merimela\_encounter\_date)

---------------------------------------

#run\_akde\_with\_data example

#Apethe\_Merimela\_encounter\_date <- ymd\_hms("2012-05-26 03:10:37")

Apethe\_Merimela\_encounter\_date <-ymd\_hms("2016-02-10 20:00:00")

#Get all data on both pairs

Apethe\_data <- movement\_data\_id\_all4 %>% filter(individual.local.identifier == "Apethe", study.id == "Wilmers")

Merimela\_data <- movement\_data\_id\_all4 %>% filter(individual.local.identifier == "Merimela", study.id == "Wilmers")

# #Split before data sets

# Apethe\_before <- Apethe\_data %>% filter(timestamp < Apethe\_Merimela\_encounter\_date)

# Merimela\_before <- Merimela\_data %>% filter(timestamp < Apethe\_Merimela\_encounter\_date)

# #Split after data sets

# Apethe\_after <- Apethe\_data %>% filter(timestamp > Apethe\_Merimela\_encounter\_date)

# Merimela\_after <- Merimela\_data %>% filter(timestamp > Apethe\_Merimela\_encounter\_date)

#Split before data sets

Apethe\_before <- Apethe\_data %>% filter(timestamp < ymd\_hms("2016-02-10 19:50:00"))

Merimela\_before <- Merimela\_data %>% filter(timestamp < ymd\_hms("2016-02-10 19:50:00"))

#Split after data sets

Apethe\_after <- Apethe\_data %>% filter(timestamp > ymd\_hms("2016-02-10 20:00:00"))

Merimela\_after <- Merimela\_data %>% filter(timestamp > ymd\_hms("2016-02-10 20:00:00"))

#Make lists to input into function (MAKE SURE IT"S NAMES)

before\_list <- list(Apethe = Apethe\_before, Merimela = Merimela\_before)

after\_list <- list(Apethe = Apethe\_after, Merimela = Merimela\_after)

#Run Function!

run\_akde\_with\_data(before\_list, after\_list, Apethe\_Merimela\_encounter\_date)