Homerange Overlaps

Kyra Bankhead

2023-07-17

In this markdown I will:

1. Convert coordinate degree data into meters and organize this data into a spatial points data frame.
2. Plot the all individual home ranges as well as HI individuals.
3. Calculate the dyadic home range overlap.

PART 1: *Spatial Points Data Frame*

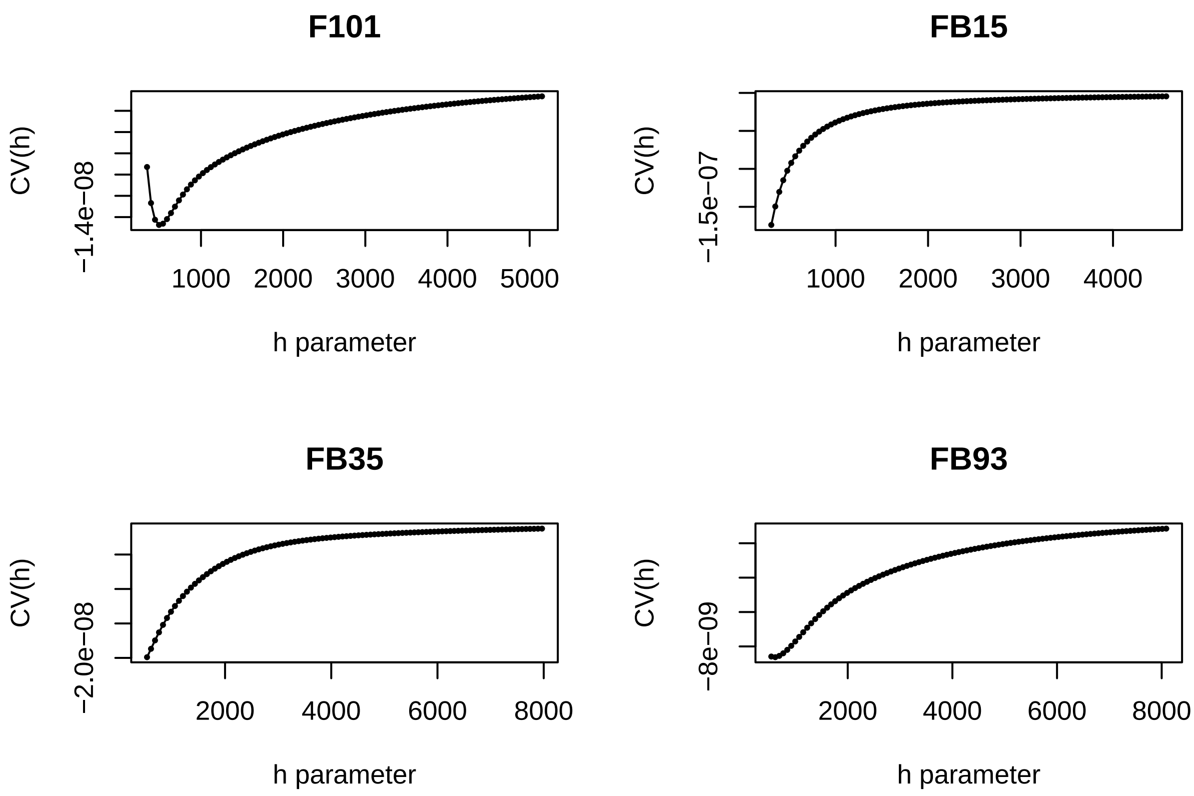
Convert coordinate degrees into meters

```
# Create a simple feature data frame (sf)
coord_data_sf <- st_as_sf(dolph.sp, coords = c("x", "y"), crs = 4326)

# UTM zone for study area
dolph.sf <- st_transform(coord_data_sf, crs = paste0("+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs
# Extract coordinates (latitude and longitude) and create new columns
dolph.sp$x <- st_coordinates(dolph.sf)[, 1]
dolph.sp$y <- st_coordinates(dolph.sf)[, 2]
# Remove two rows with NA's
dolph.sp <- dolph.sp[!is.na(dolph.sp$x) & !is.na(dolph.sp$y),]

coordinates(dolph.sp) <- c("x", "y")

# Set the initial CRS for data to WGS84 (latitude and longitude)
proj4string(dolph.sp) <- CRS( "+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs" )
```

Test which bandwidth parameter to use by investigating individuals of concern

```
# Create kernel estimates for each id
kernel.lscv <- kernelUD(dolph.sp, h = "LSCV")  # LSCV = least squares cross validation

# Get the names or IDs of the individuals that are of concern
selected_individuals <- c("F101", "F191", "F192", "FB15", "FB35", "FB41", "FB93")

# Repeat code above to calculate appropriate bandwidth for IDs of concern
```

```
concern.sp <- coord_data[, c("id", "y", "x")]
concern.sp <- subset(concern.sp, id %in% selected_individuals) # Only include selected ids

concern_sf <- st_as_sf(concern.sp, coords = c("x", "y"), crs = 4326)
concern.sf <- st_transform(concern_sf, crs = paste0("+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs")
concern.sp$x <- st_coordinates(concern.sf)[, 1]
concern.sp$y <- st_coordinates(concern.sf)[, 2]
concern.sp <- concern.sp[!is.na(concern.sp$x) & !is.na(concern.sp$y),]

coordinates(concern.sp) <- c("x", "y")

proj4string(concern.sp) <- CRS( "+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs" )

# Now recalculate kernel estimates for each id
kernel.con <- kernelUD(concern.sp, h = "LSCV")
plotLSCV(kernel.con) # it looks like a bandwidth of 1000 will be good enough
```



Calculate Kernel Density using adjusted bandwidth

PART 2: *Homerange Overlaps*

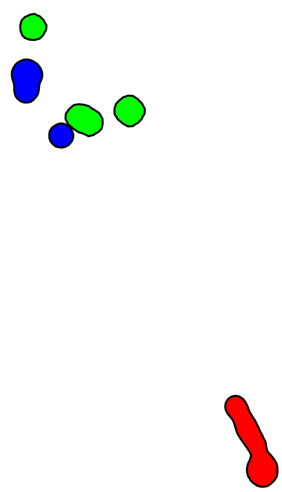
Plot all individual homerange

```
# Calculate MCPs for each HI dolphin
dolph.mcp <- mcp(dolph.sp, percent = 95)

# Plot
plot(dolph.sp, col = as.factor(dolph.sp@data$id), pch = 16, asp = 1)
plot(dolph.mcp, col = alpha(1:5, 0.5), add = TRUE)
```

Plot only HI individual home ranges using kernel estimates

```
# Find HI events among individuals
ID_HI <- subset(coord_data, subset=c(coord_data$HI == 1))
ID_HI <- ID_HI[,c('y', 'x', 'id')]

# Make sure there are at least 5 relocations
ID <- unique(ID_HI$id)
```

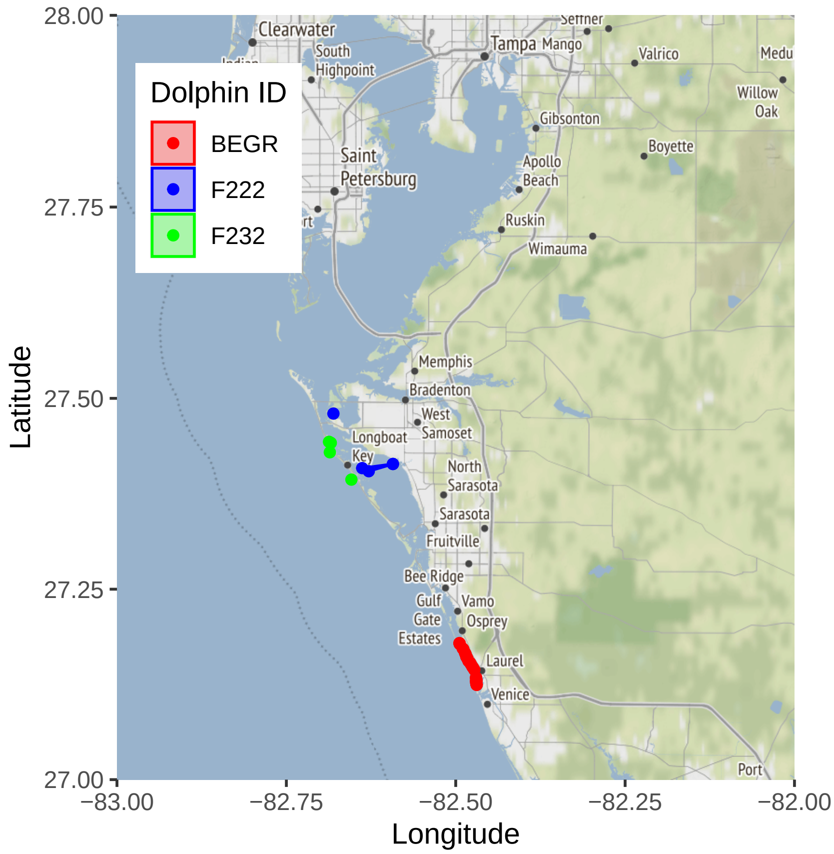
```r
obs_vect <- NULL
for (i in 1:length(ID)) {
  obs_vect[i]<- sum(ID_HI$id == ID[i])
}
sub <- data.frame(ID, obs_vect)
sub <- subset(sub, subset=c(sub$obs_vect > 4))
ID_HI <- subset(ID_HI, ID_HI$id %in% sub$ID)

# Recalculate Coordinate data
ID_HI_sf <- st_as_sf(ID_HI, coords = c("x", "y"), crs = 4326)
HI.sf <- st_transform(ID_HI_sf, crs = paste0("+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs"))
ID_HI$x <- st_coordinates(HI.sf)[, 1]
ID_HI$y <- st_coordinates(HI.sf)[, 2]

ID_HI <- ID_HI[!is.na(ID_HI$x) & !is.na(ID_HI$y),]

coordinates(ID_HI) <- c("x", "y")

proj4string(ID_HI) <- CRS( "+proj=utm +zone=17 +datum=WGS84 +units=m +no_defs" )

# Kernel estimate
HI.kern <- kernelUD(ID_HI, h = 500)
HI.kernel.poly <- getverticeshr(HI.kern, percent = 95)

# Plot kernel density
colors <- c("red", "green", "blue")
individuals <- unique(HI.kernel.poly@data$id)
## Match each individual to a color
individual_color <- colors[match(individuals, unique(HI.kernel.poly@data$id))]
## Match the color for each home range polygon
color <- individual_color[match(HI.kernel.poly@data$id, individuals)]
## Plot the home range polygons with colors
plot(HI.kernel.poly, col = color)
```



Plot only HI individual home ranges using MCP

```r
# Calculate MCPs for each HI dolphin
HI.mcp <- mcp(ID_HI, percent = 95)

# Transform the point and MCP objects.
HI.spgeo <- spTransform(ID_HI, CRS("+proj=longlat"))
HI.mcpgeo <- spTransform(HI.mcp, CRS("+proj=longlat"))

# Turn the spatial data frame of points into just a dataframe for plotting in ggmap
HI.geo <- data.frame(HI.spgeo@coords,
                          id = HI.spgeo@data$id )

# Create background map using ggmap
mybasemap <- get_stamenmap(bbox = c(left = -83, bottom = 27, right = -82, top = 28))

# Plot HI ids
ggmap(mybasemap) +
  geom_polygon(data = fortify(HI.mcpgeo),
               # Polygon layer needs to be "fortified" to add geometry to the dataframe
               aes(long, lat, colour = id, fill = id),
               alpha = 0.3) + # alpha sets the transparency
  geom_point(data = HI.geo,
             aes(x = x, y = y, colour = id))  +
```

```r
  theme(legend.position = c(0.15, 0.80)) +
  labs(x = "Longitude", y = "Latitude") +
  scale_fill_manual(name = "Dolphin ID",
                    values = c("red", "blue", "green"),
                    breaks = c("BEGR", "F222", "F232")) +
  scale_colour_manual(name = "Dolphin ID",
                      values = c("red", "blue", "green"),
                      breaks = c("BEGR", "F222", "F232"))
```



PART 3: *Dyadic Homerange Overlap*

Calculate Dyadic HRO Matrix: HRO = (Rij/Ri) * (Rij/Rj)

```r
# Get HRO
kov <- kerneloverlaphr(kernel.lscv, method="HR", lev=95)
```