

# 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]

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

Calculate Kernel Density using adjusted bandwidth

## PART 2: *Homerange Overlaps*

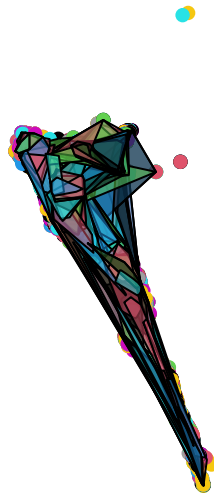
Plot all individual homerange

```

# 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 != 0))
ID_HI <- ID_HI[,c('y', 'x', 'id')]

# Make sure there are at least 5 relocations
ID <- unique(ID_HI$id)
obs_vect <- NULL
for (i in 1:length(ID)) {
  obs_vect[i] <- sum(ID_HI$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),]

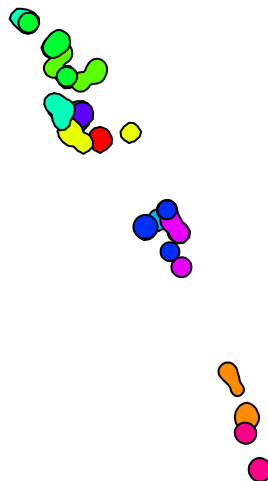
coordinates(ID_HI) <- c("x", "y")

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 <- rainbow(length(unique(ID_HI$id)))
individuals <- unique(HI.kernel.poly@data$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)

```



Look into what HI behaviors these dolphins have performed

```
# Categorize ConfHI to IDs
id_hi <- subset(coord_data, subset=c(coord_data$HI != 0))

HI_type <- as.matrix(table(id_hi$id, id_hi$HI))
HI_type <- data.frame(HI_type)
colnames(HI_type) <- c("Code", "ConfHI", "Freq")

HI <- subset(HI_type, subset=c(HI_type$Freq != 0))
HI <- subset(HI, HI$Code %in% individuals)
HI
```

##	Code	ConfHI	Freq
## 8	1651	A	4
## 19	C834	A	4
## 50	F222	A	3
## 53	F232	A	6
## 56	F241	A	4
## 64	F266	A	4
## 68	F276	A	4
## 77	FB78	A	6
## 89	SCS2	A	5
## 159	F266	B	1
## 163	F276	B	1
## 198	1651	C	1
## 209	C834	C	1
## 243	F232	C	1
## 304	C834	E	1
## 349	F266	E	1
## 392	BEGR	F	19
## 430	F222	F	1
## 487	BEGR	G	1
## 525	F222	G	1
## 528	F232	G	1
## 582	BEGR	H	1
## 623	F232	H	1
## 626	F241	H	1
## 718	F232	M	1
## 784	F113	N	6
## 874	C834	P	1

Plot only HI individual home ranges using MCP

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

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

```

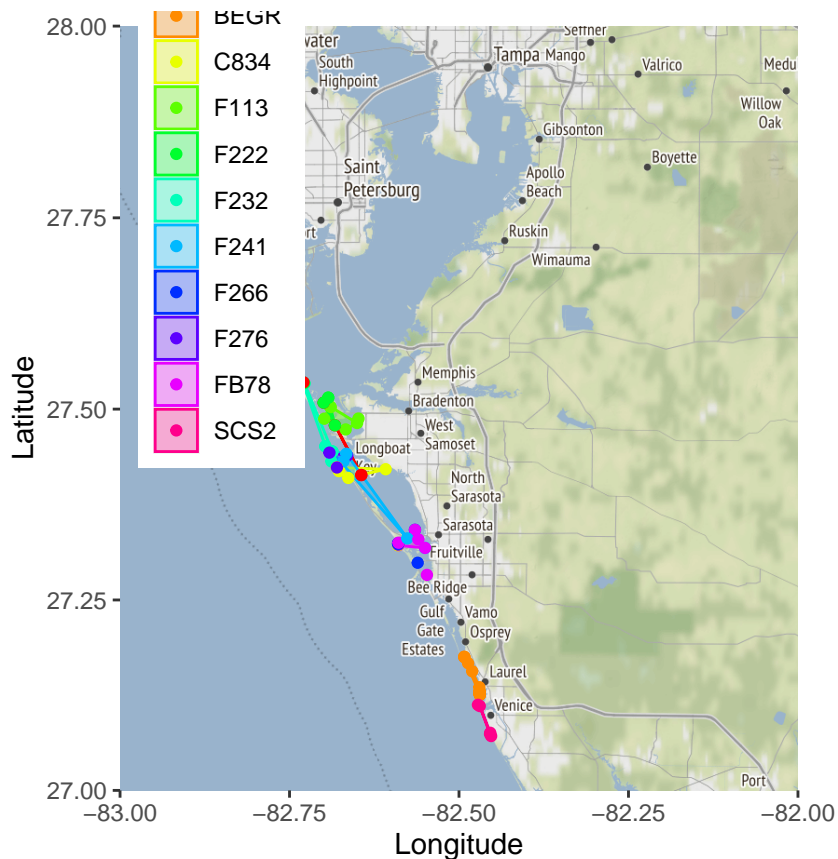
# Turn the spatial data frame of points into just a dataframe for plotting in ggmap
HI.geo <- data.frame(HI.spgeo@coords,
                     id = HI.spgeo@data$id )

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

# Plot HI ids
mymap.hr <- 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)) +
  theme(legend.position = c(0.15, 0.80)) +
  labs(x = "Longitude", y = "Latitude") +
  scale_fill_manual(name = "Dolphin ID",
                   values = colors,
                   breaks = individuals) +
  scale_colour_manual(name = "Dolphin ID",
                    values = colors,
                    breaks = individuals)

mymap.hr

```



### PART 3: *Dyadic Homerange Overlap*

Calculate Dyadic HRO Matrix:  $HRO = (R_{ij}/R_i) * (R_{ij}/R_j)$

```
# Get HRO
kernel <- kernelUD(dolph.sp, h = 500)
kov <- kerneloverlaphr(kernel, method="HR", lev=95)
```