

Study Area Maps for *Ziphius* HHMM Paper

Data Preparation

SOAR Boundaries

The data file with the SOAR outline is not being made public but can be viewed on the map figure.

```
# sf::st_layers("map_data/SOAR.gpx")
soar <- sf::st_read("map_data/SOAR.gpx",
  layer = "routes",
  quiet = TRUE)
```

Tag Location Data

```
# get data table with GPS position of each dive cycle
# note, this chunk will not run unless
#you have the (giant) full raw SMRT data files
ncpath <- "/Users/sld33/Dropbox/FBdata"
whale_meta <- read_csv("../data/smrt_hhmm_metadata.csv")
whales <- pull(whale_meta, whales)
all_locs <- list()
for (w in c(1:length(whales))){
  nc_fname = file.path(ncpath,
    paste0(whales[w], '-cal.nc'))
  all_locs[[w]] <- get_gps_data(nc_fname) |>
    mutate(whale_ID = whales[w])
}

all_locs <- bind_rows(all_locs)
tagon_locs <- all_locs |>
  filter(sec_since_tagon == 0)
end_locs <- all_locs |>
  group_by(whale_ID) |>
  slice_max(sec_since_tagon) |>
  ungroup()

whale_colors <- scico(nrow(tagon_locs), palette = "vik0")
```

```
# first element defines the start longitude, the second element the end
longitude, the third element the minimum latitude and the fourth element the
maximum latitude of the bounding box
my_bb <- c(
  c(floor(min(all_locs$longitude*10)), ceiling(max(all_locs$longitude)*10)) /
```

```
10,
  c(floor(min(all_locs$latitude*10)), ceiling(max(all_locs$latitude)*10)) / 10
)
```

GEBCO Bathymetry

To run this code, you must first download GEBCO bathymetric data for the region between 30-35 degrees N and 116 to 121 degrees W as a geoTIFF file (see: <https://www.gebco.net/data-products-gridded-bathymetry-data/gebco2025-grid>).

2025 data were used for the maps in this paper. These data are publicly available but not included in the repository.

Citation: GEBCO Compilation Group (2025) GEBCO 2025 Grid (doi:10.5285/37c52e96-24ea-67ce-e063-7086abc05f29).

```
# this version includes elevation on land
bathy_raster <- rast("map_data/GEBCO_2025_bathy/
gebco_2025_n35.0_s30.0_w-121.0_e-116.0.tif")
bathy_raster <- crop(bathy_raster, ext(my_bb))
bathy_raster <- ifel(bathy_raster >= 0, 0, bathy_raster)

# this version replaces land with NA (so will be plotted as grey on maps)
# but resolution is not high enough for zoomed-in view of island coastline
bathy_raster_masked <- ifel(bathy_raster >= 0, NA, bathy_raster)
bathy_raster_masked <- crop(bathy_raster_masked, ext(my_bb))
```

```
bathy_lims <- c(min(minmax(bathy_raster)), 0)
```

San Clemente Island Coastline

The bathymetry data has sufficient resolution to show coastlines at broad scales, but when zoomed in near SOAR, the coast of San Clemente island looks pixellated.

We obtain high-resolution data on the island's coastline from NOAA at <https://nsde.ngs.noaa.gov/>, Project ID PH7108. The shapefiles can be downloaded freely from the website but are not included in the repository.

```
land <- st_read(dsn = "map_data/PH7108/historic11.shp",
               quiet = TRUE)
land <- st_transform(land, crs = st_crs(soar))
```

Data Transformations

```
# convert the tagon locations to R sf spatial objects
# so we can plot them with geom_sf()
```

```

tagon_locs <- st_as_sf(tagon_locs,
                      coords = c("longitude",
                                  "latitude"),
                      crs = st_crs(soar))

end_locs <- st_as_sf(end_locs,
                    coords = c("longitude",
                                "latitude"),
                    crs = st_crs(soar))

all_locs <- st_as_sf(all_locs,
                    coords = c("longitude",
                                "latitude"),
                    crs = st_crs(soar))

```

```

# the deployment ID strings are too long to use as labels on the map
# so define numeric IDs which will also be included in Table 1

tagon_locs$whale_num <- factor(c(1:nrow(tagon_locs)))
end_locs$whale_num <- factor(c(1:nrow(tagon_locs)))
all_locs$whale_num <- factor(as.numeric(factor(all_locs$whale_ID)))

```

Maps

Base Map + SOAR

```

theme_set(theme_bw(base_size = 9))
wide_area_map <- ggplot() +
  geom_spatraster(data = bathy_raster_masked, show.legend = FALSE) +
  annotation_north_arrow(
    which_north = TRUE,
    height = unit(0.1, "npc"),
    width = unit(0.1, "npc"),
    pad_x = unit(0.85, "npc"),
    pad_y = unit(0.85, "npc"),
    style = north_arrow_nautical()
  ) +
  annotation_scale(
    height = unit(0.015, "npc"),
    width_hint = 0.5,
    pad_x = unit(0.07, "npc"),
    pad_y = unit(0.07, "npc"),
    text_cex = .5
  ) +
  scale_x_continuous(expand = c(0, 0)) +

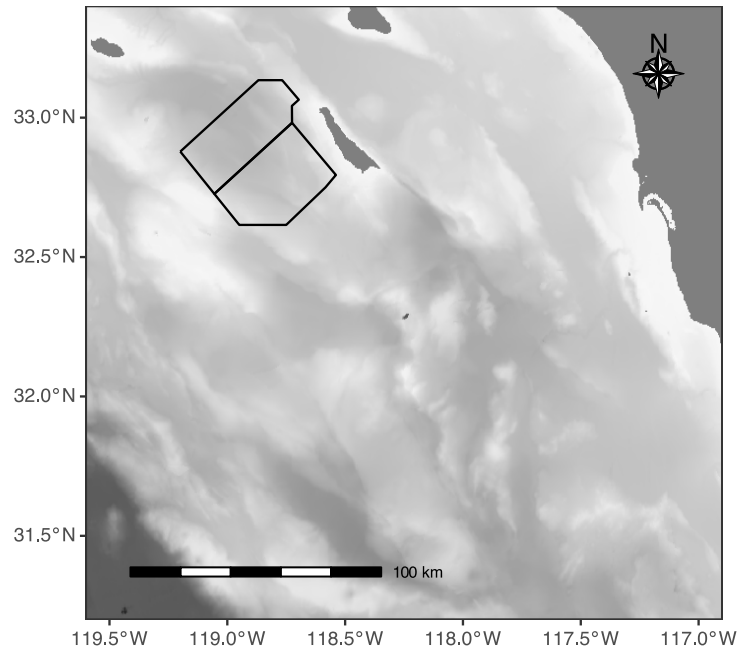
```

```

scale_y_continuous(expand = c(0, 0)) +
scale_fill_distiller(palette = "Greys",
                     direction = -1,
                     limits = bathy_lims)

wide_area_map +
  geom_sf(data = soar)

```



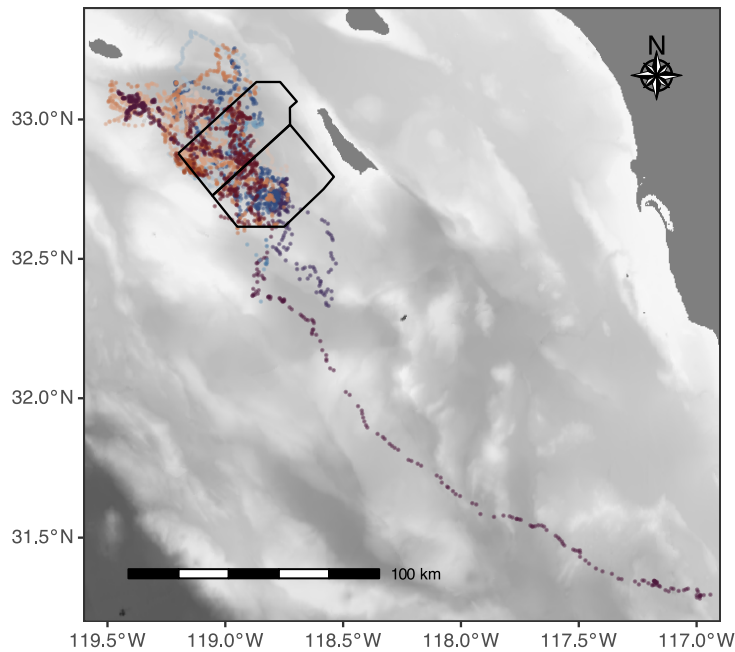
Base Map + SOAR + Whale Tracks

```

track_map <- wide_area_map +
  geom_sf(data = all_locs,
          aes(color = whale_num),
          size = 0.0075,
          alpha = 0.4,
          show.legend = FALSE) +
  scale_color_manual(values = whale_colors) +
  geom_sf(data = soar)

track_map

```



Interactive Version

If you care to see the tracks of individual whales:

```
if (knitr::is_html_output()) {
  int_track_map <- wide_area_map +
    geom_sf(data = all_locs,
            aes(color = whale_num),
            size = 0.5,
            show.legend = FALSE) +
    scale_color_manual(values = whale_colors) +
    geom_sf(data = soar)
  ggplotly(int_track_map)
}
```

Inset with Tag Deployment Locations

```
my_sm_bb <- c(-119.3, -118.2, 32.5, 33.25)
island_label <- st_sfc(st_point(x = c(-118.375, 33.05),
                                dim = "XY"),
                      crs = st_crs(soar)) |>
  st_as_sf()

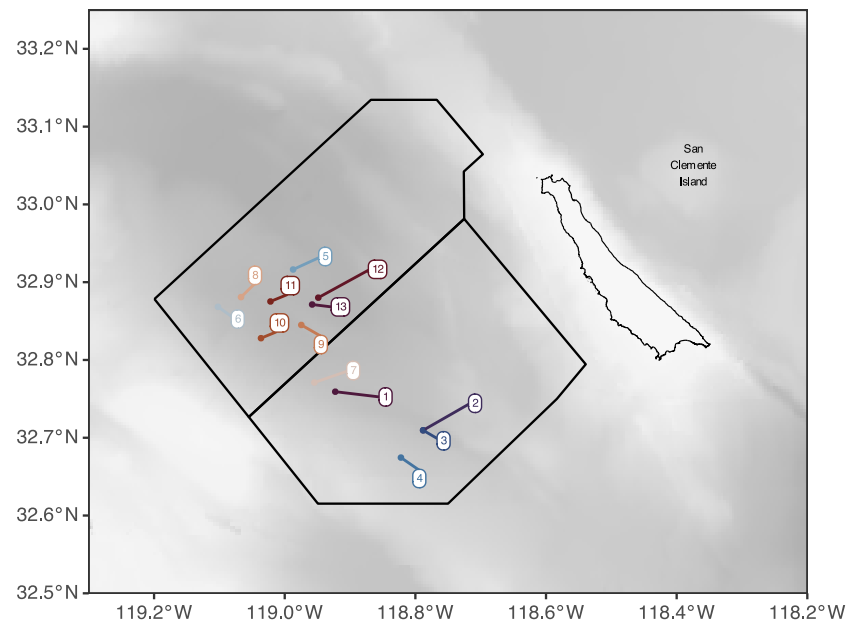
inset_map <- ggplot() +
  geom_spatraster(data = crop(bathy_raster, ext(my_sm_bb)),
                 show.legend = FALSE) +
  xlab("") + ylab("") +
```

```

geom_sf(data = land, size = 0.2) +
geom_sf(data = soar) +
geom_sf(data = tagon_locs,
        aes(color = whale_num, text = whale_ID),
        size = 0.5,
        show.legend = FALSE) +
scale_color_manual(values = whale_colors) +
scale_fill_distiller(palette = "Greys",
                    direction = -1,
                    limits = bathy_lims) +
scale_x_continuous(expand = c(0, 0)) +
scale_y_continuous(expand = c(0, 0)) +
geom_label_repel(data = tagon_locs,
                mapping = aes(label = whale_num,
                            geometry = geometry,
                            color = whale_num),
                stat = "sf_coordinates",
                size = 1.5,
                show.legend = FALSE,
                max.overlaps = 30,
                # force_pull = 0,
                # label.r = 0.05,
                nudge_x = 0.03,
                nudge_y = 0.01,
                label.padding = 0.1,
                # will this make every label attach
                # to its point w/ a line?
                min.segment.length = 0
                ) +
geom_sf_text(data = island_label,
            mapping = aes(label = "San\nClemente\nIsland"),
            size = 1.5)

```

inset_map



Interactive Version

If you would like to match up ID numbers and strings without reference to Table 1.

```
if (knitr::is_html_output()) {
  ggplotly(inset_map)
}
```

Final Figure: Base Map + Inset

```
track_map + inset_element(inset_map +
  theme(plot.background = element_rect(fill =
    "transparent", colour = NA),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()),
  left = 0.4,
  bottom = 0.21,
  right = 1,
  top = 0.7)
```

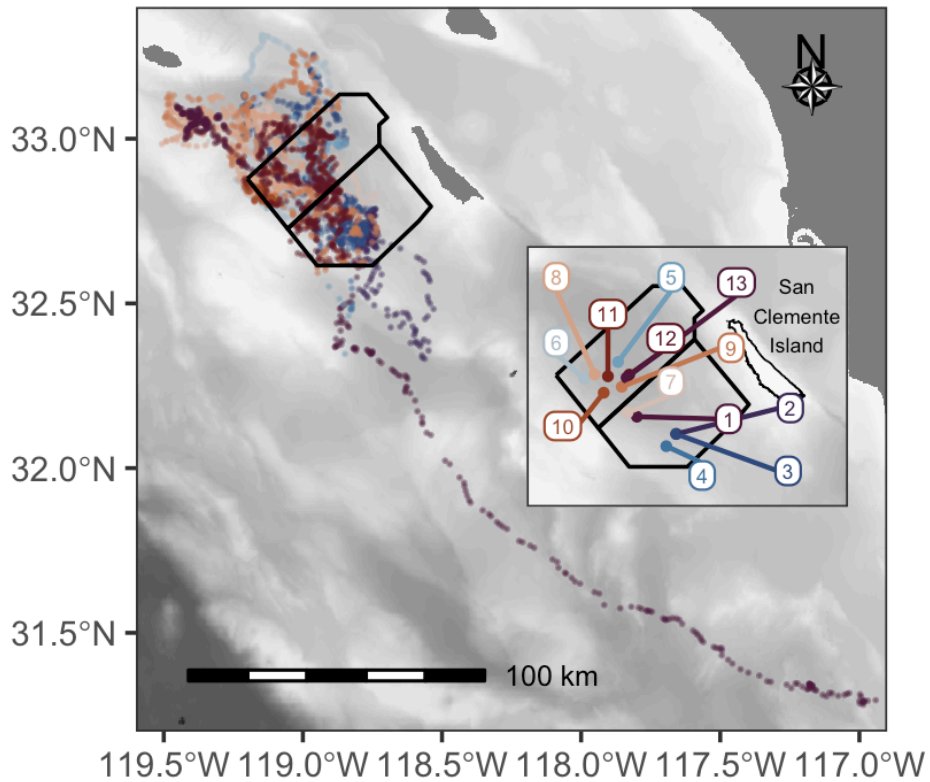


Figure 1: Map of the study area. Northern and Southern parts of the United States Navy's Southern California Anti-submarine Warfare Range (SOAR) are outlined in black. Colored dots are GPS positions from the SMRT tag data, colored by tag deployment. Inset uses the same color codings, and shows tag deployment locations labeled by tag deployment number (which can be matched with whale ID and other metadata using Table 1).