

# Sabine's Gulls outside of their main wintering areas are not necessarily vagrants

Jonatas H. F. Prado; Gabriel Canani; Pedro V. Castilho; Nicholas W. Daudt

```
rm(list = ls())
```

## Packages

### Data acquisition and wrangling

```
library(tidyverse)
library(auk)
library(robis)
```

### Thinning and kernel

```
library(spThin)
library(adehabitatHR)
```

### Mapping and spatial wrangling

```
library(sf)
library(ncdf4)
library(oceanmap)
library(raster)
library(viridis)
library(ggpattern)
```

## 01. Public databases data acquisition

The data needs to be downloaded from the website

### GBIF

```
### Read the file
gbif <-
  readxl::read_excel("./data/GBIF_raw.xlsx") %>%
  dplyr::select(Latitude = decimalLatitude,
               Longitude = decimalLongitude,
               Date = eventDate,
               Count = individualCount,
```

```

        Locality = locality,
        Country = countryCode) %>%
dplyr::filter(! is.na(Latitude))

# Add the data source
gbif$Source <- "gbif"

# Find and delete lat and long data with errors
gbif$Latitude <- as.numeric(gbif$Latitude)
gbif$Longitude <- as.numeric(gbif$Longitude)

# Create an error label and remove rows containing it
gbif <-
  gbif %>%
  # Create a error label
  dplyr::mutate(Latitude.error = ifelse(gbif$Latitude > 90 | gbif$Latitude < -90,
                                       yes = "error", no = "correct"),
               Longitude.error = ifelse(gbif$Longitude > 180 | gbif$Longitude < -180,
                                       yes = "error", no = "correct")) %>%

  # Remove rows with "error" labels
  dplyr::filter(! Latitude.error == "error" &
                ! Longitude.error == "error") %>%

  # Remove conditional columns
  dplyr::select(- c(Latitude.error, Longitude.error))

# readr::write_csv(gbif, "./data/GBIF_raw_clean.csv")

```

## eBIRD

```

# Read eBird file
ebird <-
  auk::read_ebd("./data/EBIRD_raw.txt") %>%
  dplyr::select(Latitude = latitude,
               Longitude = longitude,
               Date = observation_date,
               Count = observation_count,
               Locality = locality,
               Country = country)

# Add the data source
ebird$Source <- "ebird"

#searching for missing info
summary(is.na(ebird$Date))
summary(is.na(ebird$Latitude))

# readr::write_csv(ebird, "./data/EBIRD_raw_clean.csv")

```

## OBIS

```

# Just need to read this file, and skip for "Merge different data sources into a single data.frame"
obis <- readr::read_csv("./data/OBIS_raw.csv")

##

```

```

## Below, code used for downloading and wrangle original data set

# Download data
# obis <-
#   robis::occurrence("Xema") %>% ## Retrieved 1247 records
#   dplyr::select(Latitude = decimalLatitude,
#                 Longitude = decimalLongitude,
#                 Date = verbatimEventDate,
#                 Count = individualCount,
#                 Locality = locality,
#                 Country = waterBody)
#
# obis$Date <- as.Date(obis$Date)
#
# # Standardize to save as .csv
# obis$Locality <- gsub(";", ",", obis$Locality)
# obis$Country <- gsub(";", ",", obis$Country)
#
# # Add the data source
# obis$Source <- "obis"
#
## - - - - Last data download on April 27, 2021 - - - ##
# Save the data for reproducibility
# readr::write_csv(obis, "./data/OBIS_raw.csv")

```

Merge different data sources into a single data.frame (FINAL “RAW” DATA SET)

```

xema <- rbind(gbif, ebird, obis)

## - - - - FINAL RAW DATA SET - - - - - ##
readr::write_csv(xema, "./data/xema.csv")
## - - - - - ##

rm(list = "ebird", "gbif", "obis")

```

## 02. Thinning to avoid sampling biases

```

### The original data set was divided for the thinning procedure ('Subsets'),
### due to computational limitations

### *If* you are starting to run the code from here, read the 'xema' data set
xema <- readr::read_csv("./data/xema.csv")

xema_thin <-
  xema %>%
  dplyr::select(Latitude, Longitude) %>%
  droplevels()

```

```

xema_thin$sp <- "Xema"

# Subsets ####
xema_thin_n <- xema_thin %>% filter(Latitude > 0) %>% droplevels()

xema_thin_n2 <- xema_thin_n %>% filter(Longitude > 0) %>% droplevels()

xema_thin_n3 <- xema_thin_n %>% filter(Longitude < 0) %>% droplevels()
# summary(xema_thin_n3$Longitude)

xema_thin_n4 <- xema_thin_n3 %>% filter(Longitude < -116) %>% droplevels()
# summary(xema_thin_n4$Longitude) # median -124

xema_thin_n41 <- xema_thin_n4 %>% filter(Longitude < -124) %>% droplevels()

xema_thin_n42 <- xema_thin_n4 %>% filter(Longitude > -124) %>% droplevels()

xema_thin_n5 <- xema_thin_n3 %>% filter(Longitude > -116) %>% droplevels()
# summary(xema_thin_n5$Longitude) # median -89

xema_thin_n51 <- xema_thin_n5 %>% filter(Longitude < -89) %>% droplevels()

xema_thin_n52 <- xema_thin_n5 %>% filter(Longitude > -89) %>% droplevels()

xema_thin_s <- xema_thin %>% filter(Latitude <= 0) %>% droplevels()

# Thinning ####

#n2
sp_thin <-
  spThin::thin(xema_thin_n2,
    long.col = "Longitude",
    lat.col = "Latitude",
    spec.col = "sp",
    thin.par = 100,
    reps = 1,
    locs.thinned.list.return = T,
    write.files = F) %>%
  as.data.frame()

# Backup
# readr::write_csv(sp_thin, "./data/thinning_backup/sp_thin_n2.csv")

#n41
sp_thin2 <-
  spThin::thin(xema_thin_n41,
    long.col = "Longitude",
    lat.col = "Latitude",
    spec.col = "sp",
    thin.par = 100,
    reps = 1,
    locs.thinned.list.return = T,
    write.files = F) %>%

```

```

as.data.frame()

# Backup
# readr::write_csv(sp_thin2, "./data/thinning_backup/sp_thin_n41.csv")

#n42
sp_thin3 <-
  spThin::thin(xema_thin_n42,
               long.col = "Longitude",
               lat.col = "Latitude",
               spec.col = "sp",
               thin.par = 100,
               reps = 1,
               locs.thinned.list.return = T,
               write.files = F) %>%
as.data.frame()

# Backup
# readr::write_csv(sp_thin3, "./data/thinning_backup/sp_thin_n42.csv")

#n51
sp_thin4 <-
  spThin::thin(xema_thin_n51,
               long.col = "Longitude",
               lat.col = "Latitude",
               spec.col = "sp",
               thin.par = 100,
               reps = 1,
               locs.thinned.list.return = T,
               write.files = F) %>%
as.data.frame()

# Backup
# readr::write_csv(sp_thin4, "./data/thinning_backup/sp_thin_n51.csv")

#n52
sp_thin5 <-
  spThin::thin(xema_thin_n52,
               long.col = "Longitude",
               lat.col = "Latitude",
               spec.col = "sp",
               thin.par = 100,
               reps = 1,
               locs.thinned.list.return = T,
               write.files = F) %>%
as.data.frame()

# Backup
# readr::write_csv(sp_thin4, "./data/thinning_backup/sp_thin_n52.csv")

#s
sp_thin6 <-
  spThin::thin(xema_thin_s,

```

```

        long.col = "Longitude",
        lat.col = "Latitude",
        spec.col = "sp",
        thin.par = 100,
        reps = 1,
        locs.thinned.list.return = T,
        write.files = F) %>%
as.data.frame()

# Backup
# readr::write_csv(sp_thin4, "./data/thinning_backup/sp_thin_s.csv")

#### Merge thinned data

db_thin <- rbind(sp_thin, sp_thin2, sp_thin3,
                sp_thin4, sp_thin5, sp_thin6)

plot(y = xema_thin$Lat, x = xema_thin$Long)

db_thin$sp <- '1'

### Final thinning to a bigger radius
db_thinned <-
  spThin::thin(db_thin,
               long.col = "Longitude",
               lat.col = "Latitude",
               spec.col = "sp",
               thin.par = 200, # distance between points
               reps = 1,
               locs.thinned.list.return = T,
               write.files = F) %>%
as.data.frame()

# plot(db_thinned)

# Backup
readr::write_csv(db_thinned, "./data/thinning_backup/xema_thinned_200km.csv")

rm(list =
     "xema_thin", "xema_thin_n", "xema_thin_n2", "xema_thin_n3", "xema_thin_n4",
     "xema_thin_n41", "xema_thin_n42", "xema_thin_n5", "xema_thin_n51",
     "xema_thin_n52", "xema_thin_s", "sp_thin", "sp_thin2", "sp_thin3",
     "sp_thin4", "sp_thin5", "sp_thin6", "db_thin")

```

### 03. Kernel 75

```

### *If* you are starting to run the code from here, read the 'db_thinned' data set
db_thinned <- readr::read_csv("./data/thinning_backup/xema_thinned_200km.csv")

# Transform the "data.frame" into "SpatialPointsDataFrame"
sp::coordinates(db_thinned) = c("Longitude", "Latitude")

```

```

## Estimate UD, get volume, determine 75 vertex
xema_kernel75 <-
  db_thinned %>%
  adehabitatHR::kernelUD(h = "href", same4all = TRUE) %>%
  adehabitatHR::getvolumeUD() %>%
  adehabitatHR::getverticeshr(75)

# Save shapefile
rgdal::writeOGR(xema_kernel75, "./data/xema_kernel75_shp/xema_kernel75.shp",
  layer = "poly", driver = "ESRI Shapefile")

```

## 04. Remove points inside BirdLife and Kernel75 range maps

```

### *If* you are starting to run the code from here, read the 'xema' data set
xema <- readr::read_csv("./data/xema.csv")

# Transform 'xema' data set in a spatial feature
xema_spatial <-
  xema %>%
  dplyr::mutate(long1 = Longitude, lat1 = Latitude) %>%
  sf::st_as_sf(coords = c("long1", "lat1"), crs = 4326)

## Read Kernel75 shapefile, set CRS
kernel75 <-
  sf::read_sf("./data/xema_kernel75_shp/xema_kernel75.shp") %>%
  sf::st_set_crs(4326)

## Read BirdLife shapefile, already set in CRS 4326
birdlife_raw <-
  sf::read_sf("./data/birdlife_shp/Xema_sabini.shp")

# Split it's rows in different features
birdlife1 <- birdlife_raw[1,] # 'SEASON == 2' Breeding season (br)
birdlife2 <- birdlife_raw[2,] # 'SEASON == 2' Breeding season (br)
birdlife3 <- birdlife_raw[4,] # 'SEASON == 3' Non-breeding season (nbr)
birdlife_passage <-
  birdlife_raw[3,] %>% # 'SEASON == 4' Passage
  sf::st_buffer(dist = 0) # To correct an internal bug from the feature

# Extend them with a small buffer around
birdlife_passage <-
  sf::st_buffer(birdlife_passage, dist = 1) # dist = decimal degrees

birdlife_br_nbr <-
  sf::st_union(birdlife1, birdlife2) %>%
  sf::st_union(., birdlife3) %>%
  sf::st_buffer(dist = 0.5) # dist = decimal degrees

## Filter out records inside these polygons
xema_spatial <-

```

```

sf::st_difference(xema_spatial, kernel75)

xema_spatial <-
  sf::st_difference(xema_spatial, birdlife_br_nbr)

xema_spatial <-
  sf::st_difference(xema_spatial, birdlife_passage)

## Check
# mapview::mapview(kernel75) + birdlife_br_nbr + birdlife_passage + xema_spatial

# Transform 'data' to data.frame
xema <-
  as.data.frame(xema_spatial) %>%
  dplyr::select(Latitude, Longitude, Date, Count,
               Locality, Country, Source)

# Backup
# readr::write_csv(xema, "./data/xema_after_spatial_filters1.csv")

rm(list = "kernel75", "birdlife_raw", "birdlife1", "birdlife2", "birdlife3",
     "birdlife_passage", "birdlife_br_nbr", "xema_spatial")

```

## 05. Remove records considered in usual areas and close to breeding areas

```

### *If* you are starting to run the code from here, read the 'xema' data set
xema <- readr::read_csv("./data/xema_after_spatial_filters1.csv")

# Records above 60N latitude
xema <- xema %>% filter(Latitude < 60) %>% droplevels() # Removed 45 records

# Records above 20S latitude in South America and West Africa (usual passage area)
xema <-
  xema %>%
  # Create a flag, in/out for West South America
  dplyr::mutate(sam_out = ifelse(Latitude >= -20 & Latitude <= 0 &
                               Longitude <= -50 & Longitude >= -80,
                               "in", "out")) %>%
  # Create a flag, in/out for West Africa
  dplyr::mutate(afr_out = ifelse(Latitude <= 7 & Latitude >= -2 &
                               Longitude >= -4.5 & Longitude <= 9.5,
                               "in", "out")) %>%
  # Remove records 'in'
  dplyr::filter(! sam_out == "in") %>% # Removed 277 records
  dplyr::filter(! afr_out == "in") %>% # Removed 19 records
  # Remove columns
  dplyr::select(- c(sam_out, afr_out)) %>%
  # Remove two records in the Bering Sea (assuming they are in the usual passage area)
  dplyr::filter(! Locality %in% "OD0818010-70") %>%

```



```
dplyr::filter(! Locality %in% "Zhemchug Canyon")
```

```
# Backup
```

```
# readr::write_csv(xema, "./data/xema_after_spatial_filters2.csv")
```

## 06. Add manually records from literature

```
### *If* you are starting to run the code from here, read the 'xema' data set  
xema <- readr::read_csv("./data/xema_after_spatial_filters2.csv")
```

```
# Records added from references below:
```

```
# Ash 1983 (locality of 'Hal Hambo' not found, so assumed to be on the capital Mogadishu, following the
```

```
# Andrew 1985 (longitude corrected to 100-E, instead of 101-E [on land]);
```

```
# Parrini & Carvalho 2009; Sreenivasan et al. 2013;
```

```
# Lees et al. 2014; Allport 2018; Lambert 2005 (4 records); Sinclair 1981;
```

```
# two records reported in this study in the Supplementary Material
```

```
df_input <- data.frame(  
  Latitude = as.numeric(c(2.018, 2.016, -32.217, 10.580,  
    -2.040, -25.936, -26.075,  
    -21.066, -22.383,  
    -25.600, -46.800,  
    -28.481, -31.675)),  
  Longitude = as.numeric(c(45.333, 100.500, -52.189, 75.983,  
    -44.083, 32.979, 32.766,  
    35.733, 35.783,  
    32.950, 37.698,  
    -48.681, -49.793)),  
  Date = as.Date(c("1981-05-08", "1984-10-22", "2009-11-16", "2013-05-03",  
    "2013-08-24", "2015-05-10", "1981-02-18",  
    "1988-03-23", "1988-04-01",  
    "1988-04-01", "1979-02-10",  
    "2017-03-16", "2017-06-08")),  
  Count = as.numeric(c(1, 1, 1, 1,  
    1, 1, 1,  
    1, 1,  
    1, 3,  
    1, 1)),  
  Locality = as.character(c("Hal Hambo", "NA", "Cassino beach", "Puthankadapuram beach",  
    "Praia Raposa", "north of Inhaca Island", "Delagoa Bay",  
    "northeast of Bazamto Island", "southeast of Ponta Sao Sebastiao",  
    "Delagoa Bay", "Marion Island",  
    "Praia da Vila", "off Rio Grande do Sul State")),  
  Country = as.character(c("Somalia", "Sumatra", "Brazil", "India",  
    "Brazil", "Mozambique", "Mozambique",  
    "Mozambique", "Mozambique",  
    "Mozambique", "South Africa",  
    "Brazil", "Brazil")),  
  Source = as.character(rep("literature", times = 13))  
)
```

```
xema <-
  dplyr::bind_rows(df_input, xema) %>%
  dplyr::arrange(Date)

## Check
# xema_spatial <-
#   xema %>%
#   dplyr::mutate(long1 = Longitude, lat1 = Latitude) %>%
#   sf::st_as_sf(coords = c("long1", "lat1"), crs = 4326)
#
# mapview::mapview(xema_spatial)

# Backup
# readr::write_csv(xema, "./data/xema_after_spatial_filters3.csv")

rm(list = "df_input")
```

## 07. Assign in/out from high productive and/or fishing areas for each record

```
### *If* you are starting to run the code from here, read the 'xema' data set
xema <- readr::read_csv("./data/xema_after_spatial_filters3.csv")

# After the local- and global relevance literature search for each area/record
# See main text and Table for more information

xema <-
  xema %>%
  dplyr::mutate(high_chl_fish = "in") %>%
  dplyr::mutate(high_chl_fish = ifelse(Latitude == 32.50045 & Longitude == -165.49162 |
    Latitude == 51.43110 & Longitude == 60.14530 |
    Latitude == -33.41666 & Longitude == 145.91667 |
    Locality == "Bremer Bay" |
    Locality == "Bremer Bay Boat Harbour" |
    Locality == "Bremer Bay boat harbour" |
    Locality == "Bremer Bay Fisheries Beach" |
    Locality == "AU-WA-Swarbrick Road (-34.4256,119.4000)" |
    Locality == "Fishery Beach, Bremer Bay AU-WA (-34.4261,119.3999)" |
    Locality == "Coconut Well (Broome)" |
    Locality == "Munkyarra Wetland" |
    Locality == "Lake Brewster",
    "out", high_chl_fish)) %>%
  dplyr::mutate(high_chl_fish = tidyr::replace_na(high_chl_fish, "in"))

## Check
# xema_spatial <-
#   xema %>%
#   dplyr::mutate(long1 = Longitude, lat1 = Latitude) %>%
#   sf::st_as_sf(coords = c("long1", "lat1"), crs = 4326)
#
```

```
# mapview::mapview(xema_spatial, zcol = "high_chl_fish")

# Backup
# readr::write_csv(xema, "./data/xema_after_spatial_filters4.csv")
```

## 08. Check for duplicates and finish wrangling (FINAL DATA SET - UNUSUAL RECORDS)

```
### Records with same coordinate and date were considered duplicated and
### removed of the dataset

### *If* you are starting to run the code from here, read the 'xema' data set
xema <- readr::read_csv("./data/xema_after_spatial_filters4.csv")

xema <-
  xema %>%
  dplyr::mutate(Latitude = round(Latitude, digits = 2),
               Longitude = round(Longitude, digits = 2)) %>%
  dplyr::group_by(Date, Latitude, Longitude) %>%
  dplyr::mutate(num_dups = n(),
               dup_id = dplyr::row_number()) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(is_duplicated = dup_id > 1)

xema_unusual <-
  xema %>%
  dplyr::filter(! is_duplicated == TRUE) %>%
  dplyr::select(- c(num_dups, dup_id, is_duplicated))

## - - - - UNUSUAL RECORDS - - - - -##
readr::write_csv(xema_unusual, "./data/xema_unusual.csv")
## - - - - -##

rm(list = "xema")
```

## 09. Unusual Xema - all records

```
### *If* you are starting to run the code from here, read the 'xema' data set
xema_unusual <- readr::read_csv("./data/xema_unusual.csv")

## Load world map
world <- ggplot2::map_data("world")

## Load shapefile from BirdLife again and wrangle it
birdlife_raw <-
  sf::read_sf("./data/birdlife_shp/Xema_sabini.shp")
```

```

# Split it's rows in different features
birdlife1 <- birdlife_raw[1,] # 'SEASON == 2' Breeding season (br)
birdlife2 <- birdlife_raw[2,] # 'SEASON == 2' Breeding season (br)
birdlife_nbr <- birdlife_raw[4,] # 'SEASON == 3' Non-breeding season (nbr)

# Create a small buffer around this shapefiles, as used in our filter process
birdlife_br <-
  sf::st_union(birdlife1, birdlife2) %>%
  sf::st_buffer(dist = 0.5)

birdlife_nbr <-
  birdlife_nbr %>%
  sf::st_buffer(dist = 0.5)

## Worldwide unusual Xema map
worldwide_xema_unusual <-
  ggplot() +
  geom_polygon(data = world, aes(x = long, y = lat,
                                group = group), col=NA, lwd=1, fill = "lightgray") +
  ggplot2::geom_sf(data=birdlife_br, fill="darkgray") +
  ggplot2::geom_sf(data = birdlife_nbr, fill = "black") +
  geom_point(data = xema_unusual,
             aes(x = Longitude, y = Latitude,
                 colour = factor(high_chl_fish),
                 fill = factor(high_chl_fish),
                 shape = factor(high_chl_fish)),
             size = 3) +
  scale_fill_manual(values = c("in" = "white",
                               "out" = "white")) +
  scale_colour_manual(values = c("in" = "black",
                                 "out" = "black")) +
  scale_shape_manual(values = c("in" = 21,
                                "out" = 24)) +
  ggplot2::theme_bw() +
  ggplot2::xlab("") + ggplot2::ylab("") +
  ggplot2::theme(legend.position = "none")

ggplot2::ggsave(worldwide_xema_unusual,
                 filename = "./maps/xema_unusual_all.tiff",
                 height = 210 , width = 297, units = "mm", dpi = 300)

```

## 10. Chlorophyll A maps

```

### *If* you are starting to run the code from here, read the 'xema' data set
xema_unusual <- readr::read_csv("./data/xema_unusual.csv")

## Transform 'longitude' for a 0-360 system
xema_lon360 <-
  xema_unusual %>%
  dplyr::mutate(Lon360 = ifelse(Longitude > 0, Longitude,
                               ifelse(Longitude < 0, 360 + Longitude,

```

```

"error")))

# For ease the plot, split the data based on 'in/out' productive/fisheries areas
xema_productive <-
  xema_lon360 %>% dplyr::filter(high_chl_fish == "in")

xema_non_productive <-
  xema_lon360 %>% dplyr::filter(high_chl_fish == "out")

### Open 'netCDF' file with chlorophyll data and transform it into raster
chl_data <- ncdf4::nc_open("./data/A20021852017273.L3m_CU_CHL_chlor_a_4km.nc")
chl_raster <- oceanmap::nc2raster(chl_data, "chlora",
  lonname = "lon", latname = "lat",
  date = FALSE)

chl_360 <- raster::shift(raster::rotate(raster::shift(chl_raster, 180)), 180)
chl_360 <- raster::flip(chl_360, direction = 'y')

## Black-and-white color pallet
cpal <- grDevices::gray.colors (n = 100, start = 0, end = 1,alpha = 1)

## South America (crop, plot, save)
crop_SAmerica <- raster::crop(chl_360, raster::extent(c(270, 335, -58, 15)))

tiff(file="chl_SAmerica.tiff",compression = "lzw",width=600,height=600)

oceanmap::v(crop_SAmerica, pal = cpal, zlim = c(0,1),
  cb.xlab = expression("Chlorophyll-a (mg m-3)"),
  cbpos = "r",
  grid = TRUE,
  bwd = 0.01,
  replace.na = FALSE,
  Save = FALSE,
  show.colorbar = F,
  axeslabels = F,
  subplot= T)

points(xema_non_productive$Lon360, xema_non_productive$Latitude,
  pch = 24, col = "white", bg = "black", cex = 3)
points(xema_productive$Lon360, xema_productive$Latitude,
  pch = 21, col = "black", bg = "white", cex = 3)

dev.off()

## Indian Ocean, Antarctica (crop, plot, save)
crop_IndianOc <- raster::crop(chl_360, raster::extent(c(20, 80, -75, 30)))

tiff(file="chl_IndianOcean.tiff",compression = "lzw",width=450,height=600)

oceanmap::v(crop_IndianOc, pal = cpal, zlim = c(0,1),
  cb.xlab = expression("Chlorophyll-a (mg m-3)"), cbpos = "r",
  grid = TRUE, bwd = 0.01,
  replace.na = FALSE,

```

```

    Save = FALSE,
    show.colorbar= F,
    axeslabels = F,
    subplot=T)

points(xema_non_productive$Lon360, xema_non_productive$Latitude,
       pch = 24, col = "white", bg = "black", cex = 2)
points(xema_productive$Lon360, xema_productive$Latitude,
       pch = 21, col = "black", bg = "white", cex = 3)

dev.off()

## Pacific Ocean (crop, plot, save)deviceDim()
crop_PacificOc <- raster::crop(chl_360, raster::extent(c(90, 190, -45, 70)))

tiff(file="chl_PacificOcean.tiff",compression = "lzw",width=700,height=600)

oceanmap::v(crop_PacificOc, pal = cpal, zlim = c(0,1),
            cb.xlab = expression("Chlorophyll-a (mg m-3)"),
            cbpos = "r",
            grid = T, bwd = 0.01,
            replace.na = FALSE,
            Save = FALSE,
            axeslabels = F,
            subplot=T)

xema_non_productive_f <-
  xema_non_productive %>%
  dplyr::filter(Lon360 < 180)

xema_productive_f <-
  xema_productive %>%
  dplyr::mutate (Lon360 = as.numeric(Lon360)) %>%
  dplyr::filter(Lon360 > 90)

points(xema_non_productive_f$Lon360, xema_non_productive_f$Latitude,
       pch = 24, col = "white", bg = "black", cex = 2)
points(xema_productive_f$Lon360, xema_productive_f$Latitude,
       pch = 21, col = "black", bg = "white", cex = 3)

dev.off()

## Final Figure 2 plate was built under
## nome-do-programa software, using the above saved files

## Zoom-in Mozambique
crop_Mozambique <- raster::crop(chl_360, raster::extent(c(25, 50, -40, -15)))

oceanmap::v(crop_Mozambique, pal = cpal, zlim = c(0,1),
            cb.xlab = expression("Chlorophyll-a (mg m-3)"), cbpos = "r",
            grid = TRUE, bwd = 0.01,
            replace.na = FALSE,
            Save = FALSE)

```

```

points(xema_non_productive$Lon360, xema_non_productive$Latitude,
       pch = 24, col = "white", bg = "black", cex = 2)
points(xema_productive$Lon360, xema_productive$Latitude,
       pch = 21, col = "black", bg = "green", cex = 2)

grDevices::savePlot(filename = "./maps/chl_Mozambique.tiff", type = "tiff")

## Zoom-in ~Somalia and north Indian Ocean
crop_Somalia_nIndian <- raster::crop(chl_360, raster::extent(c(30, 80, -5, 30)))

oceanmap::v(crop_Somalia_nIndian, pal = cpal, zlim = c(0,1),
            cb.xlab = expression("Chlorophyll-a (mg m-3)"), cbpos = "r",
            grid = TRUE, bwd = 0.01,
            replace.na = FALSE,
            Save = FALSE)

points(xema_non_productive$Lon360, xema_non_productive$Latitude,
       pch = 24, col = "white", bg = "black", cex = 2)
points(xema_productive$Lon360, xema_productive$Latitude,
       pch = 21, col = "black", bg = "green", cex = 2)

grDevices::savePlot(filename = "./maps/chl_Somalia_and_northIndian.tiff", type = "tiff")

```

## 11. Supplementary Material

### Raw public data databases

```

## Load raw data
gbif <- readr::read_csv("./data/GBIF_raw_clean.csv")
ebird <- readr::read_csv("./data/EBIRD_raw_clean.csv")
obis <- readr::read_csv("./data/OBIS_raw.csv")

## Load world map
world <- ggplot2::map_data("world")

## GBIF
gbif_xema <-
  ggplot2::ggplot() +
  ggplot2::geom_map(data = world, map = world,
                   aes(x = long, y = lat, map_id = region),
                   color = "black", fill = "lightgray", size = 0.1) +
  ggplot2::geom_point(data = gbif,
                      aes(x = Longitude, y = Latitude,
                          color = "red")) +
  ggplot2::theme_bw() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  ggplot2::theme(legend.position = "none")

ggplot2::ggsave(gbif_xema,
                 filename = "./maps/gbif_xema.pdf",

```

```

        height = 210 , width = 297, units = "mm", dpi = 300)

## eBird
ebird_xema <-
  ggplot2::ggplot() +
  ggplot2::geom_map(data = world, map = world,
                    aes(x = long, y = lat, map_id = region),
                    color = "black", fill = "lightgray", size = 0.1) +
  ggplot2::geom_point(data = ebird,
                      aes(x = Longitude, y = Latitude,
                          color = "blue")) +
  ggplot2::scale_color_manual(values = "blue") +
  ggplot2::theme_bw() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  ggplot2::theme(legend.position = "none")

ggplot2::ggsave(ebird_xema,
                filename = "./maps/ebird_xema.pdf",
                height = 210 , width = 297, units = "mm", dpi = 300)

## OBIS
obis_xema <-
  ggplot2::ggplot() +
  ggplot2::geom_map(data = world, map = world,
                    aes(x = long, y = lat, map_id = region),
                    color = "black", fill = "lightgray", size = 0.1) +
  ggplot2::geom_point(data = obis,
                      aes(x = Longitude, y = Latitude,
                          color = "green")) +
  ggplot2::scale_color_manual(values = "green") +
  ggplot2::theme_bw() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  ggplot2::theme(legend.position = "none")

ggplot2::ggsave(obis_xema,
                filename = "./maps/obis_xema.pdf",
                height = 210 , width = 297, units = "mm", dpi = 300)

```

## Kernel and thinned records

```

## Load thinned database and kernel75 shapefile
db_thinned <- readr::read_csv("./data/thinning_backup/xema_thinned_200km.csv")
kernel75 <- sf::read_sf("./data/xema_kernel75_shp/xema_kernel75.shp")

## Load world map
world <- ggplot2::map_data("world")

kernel_thinned_xema <-
  ggplot2::ggplot() +
  ggplot2::geom_map(data = world, map = world,
                    aes(x = long, y = lat, map_id = region),

```



```
        color = "black", fill = "lightgray", size = 0.1) +
ggplot2::geom_point(data = db_thinned,
                    aes(x = Longitude, y = Latitude)) +
ggplot2::geom_sf(data = kernel75, fill = "red", alpha = 0.5) +
ggplot2::theme_bw() +
ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
ggplot2::theme(legend.position = "none")

ggplot2::ggsave(kernel_thinned_xema,
                 filename = "./maps/kernel_thinned_xema.pdf",
                 height = 210 , width = 297, units = "mm", dpi = 300)
```

**end**