#1B Create the figures associated with the jump analysis

Nadege Belouard* Isabella G. Smith†

2024-08-13

Contents

Aim and setup

This vignette computes all the figures included in the manuscript that is associated with the `jumpID` package. This vignette requires loading data frames generated in the first vignette of this package.

*iEco lab at Temple University, Ecobio lab at the University of Rennes, nadege.belouard@gmail.com
†iEco lab at Temple University

```
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
library(ggplot2)
library(cowplot)
library(jumpID)
```

Load files generated in the previous vignette

```
slf <- read.csv(file.path(here::here(), "data", "lyde_data_v2", "lyde.csv"), h=T)
grid_data <- read.csv(file.path(here::here(), "exported-data", "grid_data.csv"), h=T)
centroid <- data.frame(longitude_rounded = -75.675340, latitude_rounded = 40.415240)

Jumps <- read.csv(file.path(here::here(), "exported-data", "jumps.csv"))
Jump_clusters <- read.csv(file.path(here::here(), "exported-data", "jump_clusters.csv"))
Thresholds <- read.csv(file.path(here::here(), "exported-data", "thresholds.csv"))
secDiffusion <- read.csv(here::here("exported-data", "secdiffusion.csv"))
```

Prepare the states background map

```
# get a simple feature objects for states
states <- sf::st_as_sf(maps::map("state", plot = FALSE, fill = TRUE)) %>%
  sf::st_transform(crs = 4326)
```

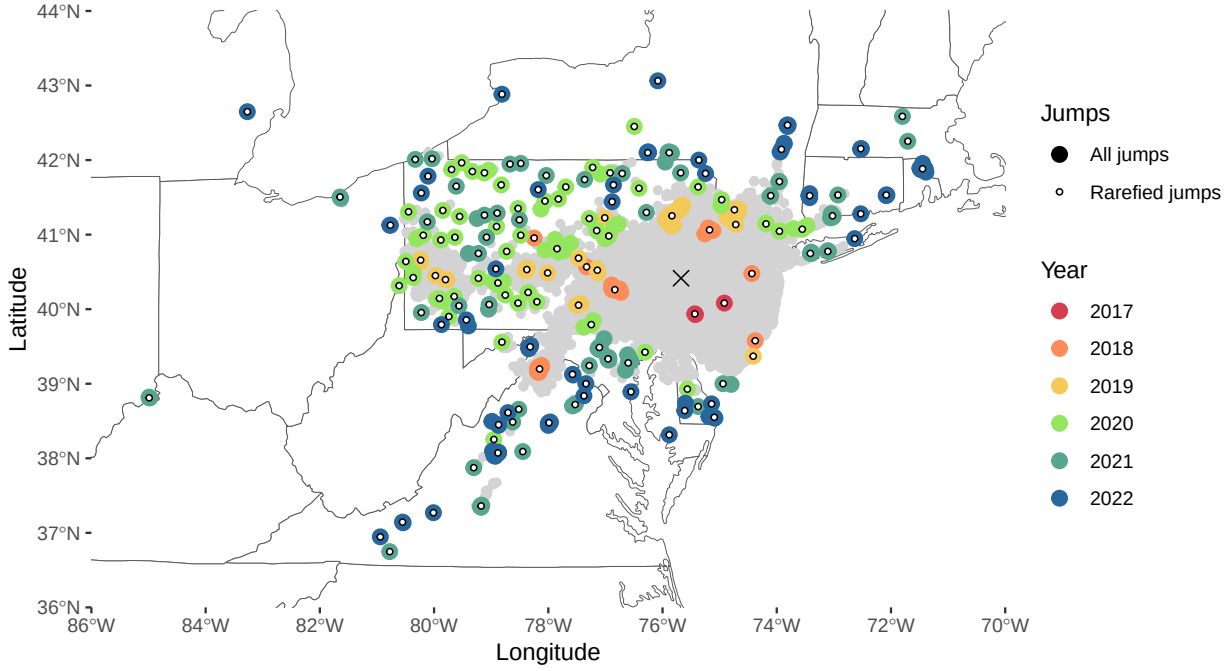
Figure 1: Faceted jump map, barplot & distance

1A: jump map

Map the position of jumps and identify jump clusters per year

```
map_rarefied <- ggplot(data = states) +
  geom_sf(data = states, fill = "white") +
  # positive points
  geom_point(data = grid_data %>% filter(established == TRUE),
             aes(x = longitude, y = latitude), col = "lightgrey") +
  # introduction point
  annotate("point", x = -75.675340, y = 40.415240,
           col = "black", shape = 4, size = 3) +
```

```
  # all jumps
  geom_point(data = Jumps,
             aes(x = longitude, y = latitude, col = as.factor(year),
             shape = "All jumps"), size = 3) +
  # jump clusters
  geom_point(data = Jump_clusters,
             aes(x = longitude, y = latitude, shape = "Rarefied jumps"),
             col = "black", fill = "white", size = 1) +

  scale_color_manual(name = "Year",
                     values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e"),
                     labels = c("2017", "2018", "2019", "2020", "2021", "2022")) +
  scale_shape_manual(name = "Jumps",
                     values = c("All jumps" = 19, "Rarefied jumps" = 21)) +
  coord_sf(xlim = c(-86, -70), ylim = c(36, 44), expand = FALSE) +
  labs(x = "Longitude", y = "Latitude") +
  theme(panel.background = element_rect(fill = "white"))

map_rarefied
```




```
ggsave(file.path(here::here(), "figures", "1A. jumps_map.jpg"),
       map_rarefied, height = 8, width = 8)
```

1B: number of jumps per year bar plot

Count how many jumps there are per year

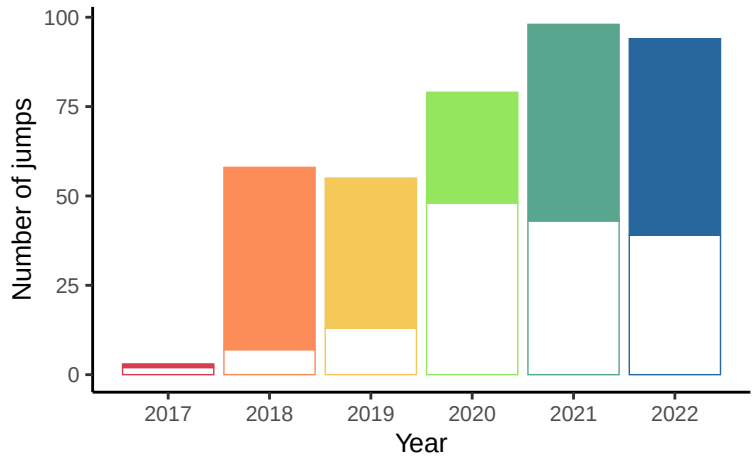
```
Jump_clusters %<>% mutate(Type = "Rarefied jumps")
Jumps %<>% mutate(Type = "All jumps")

Clusters_year <- Jump_clusters %>%
  group_by(year, Type) %>%
  summarise(n = n())
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
Jumps_year <- Jumps %>%
  group_by(year, Type) %>%
  summarise(n = n()) %>%
  left_join(Clusters_year, by = "year") %>%
  mutate(n = n.x - n.y) %>%
  rename(Type = Type.x)
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
Jumps_total <- bind_rows(Clusters_year, Jumps_year %>% select(year, Type, n))

jumps_plot <- ggplot() +
  geom_bar(data = Jumps_total,
           aes(x = year, y = n, fill = as.factor(year), col = as.factor(year),
               group = Type, alpha = Type),
           stat = "identity", lwd = .25, show.legend = F) +
  scale_fill_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  scale_color_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  scale_alpha_manual(values = c(1, 0)) +
  xlab("Year") + ylab("Number of jumps") +
  theme_classic() +
  scale_x_continuous(breaks = seq(2017, 2022, by = 1)) +
  theme(text = element_text(size = 10))

jumps_plot
```

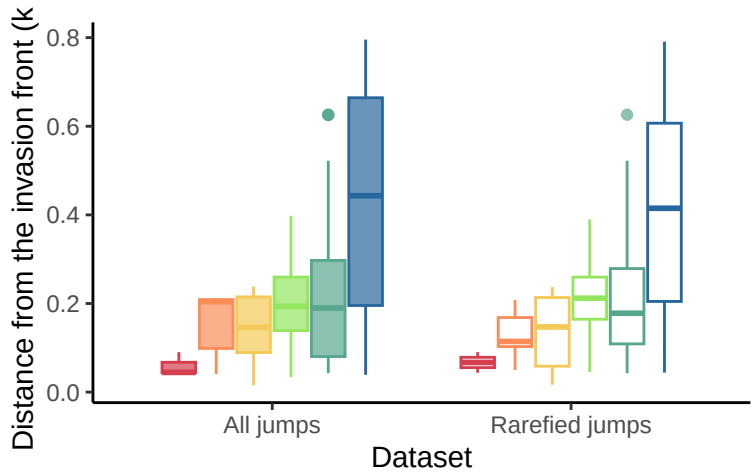



```
ggsave(file.path(here::here(), "figures", "1B. number of jumps.jpg"),
       jumps_plot, height = 2.5, width = 4)
```

1C: distance of jumps box plot

Show the distance between the invasion front and jumps every year. This variable is stored in the DistToSLF column.

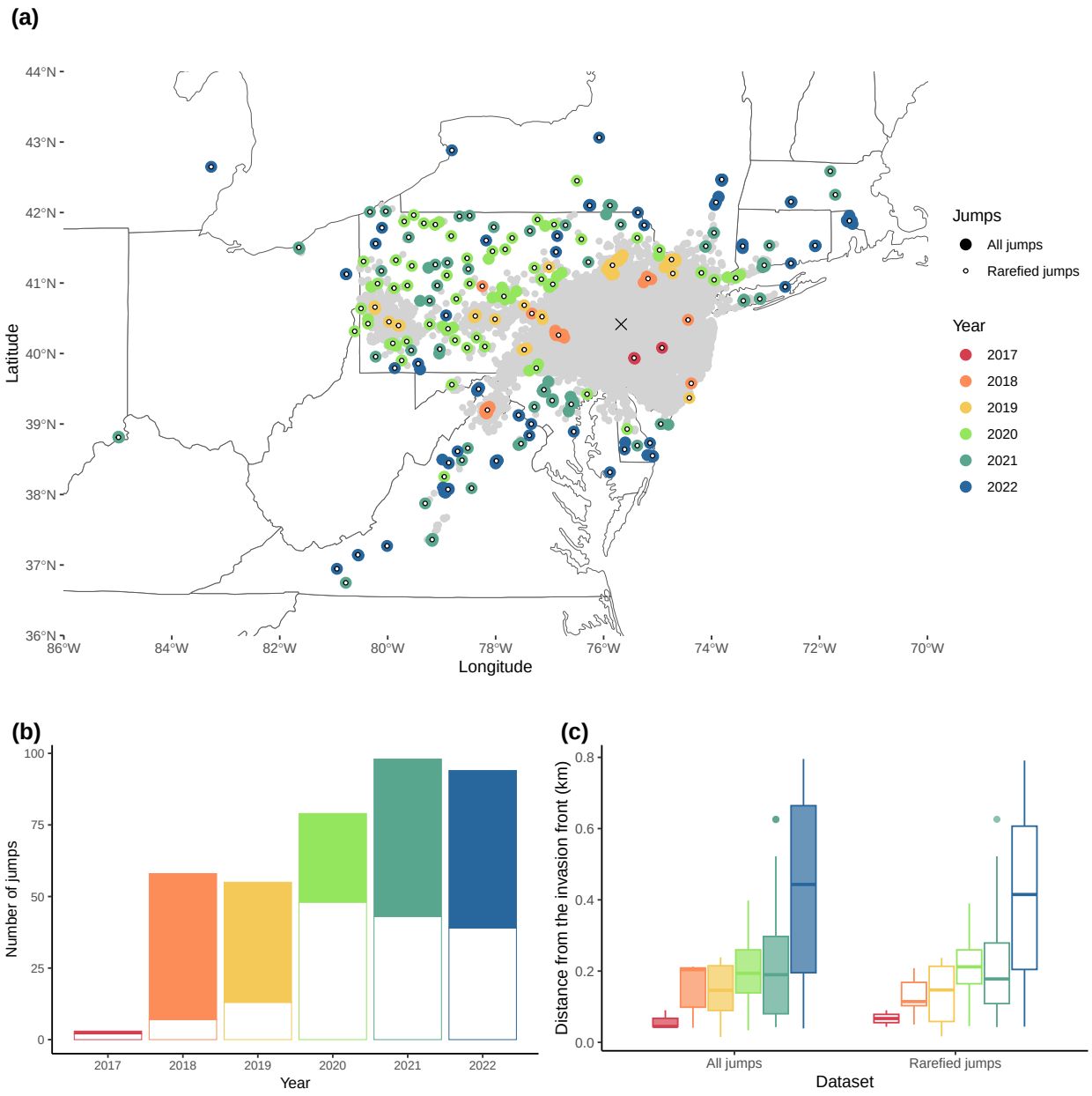
```
MeanDist_jump <- ggplot() +
  geom_boxplot(data = Jumps, show.legend = F,
               aes(x = Type, y = DistToSLF/1000,
                   col = as.factor(year),
                   fill = as.factor(year)),
               alpha = 0.7) +
  geom_boxplot(data = Jump_clusters, show.legend = F,
               aes(x = Type, y = DistToSLF/1000,
                   col = as.factor(year),
                   fill = as.factor(year)),
               alpha = 0, outlier.alpha = 0.7) +

  scale_color_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  scale_fill_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  labs(x = "Dataset", y = "Distance from the invasion front (km)") +
  theme_classic()

MeanDist_jump
```




```
ggsave(file.path(here::here(), "figures", "1C. jump_distances.jpg"),
       MeanDist_jump, height = 2.5, width = 4)
```

Assemble figure 1ABC

```
fig1 <- ggdraw() +
  draw_plot(map_rarefied, x = 0, y = .33, width = 1, height = .66) +
  draw_plot(jumps_plot, 0, 0, .5, .33) +
  draw_plot(MeanDist_jump, .5, 0, .5, .33) +
  draw_plot_label(c("(a)", "(b)", "(c)"), c(0, 0, 0.5), c(1, 0.35, 0.35), size = 15) +
  theme(plot.background = element_rect(fill="#FFFFFF", color = NA))

fig1
```




```
ggsave(file.path(here::here(), "figures", "1. jump_description.jpg"),
       fig1, height = 10, width = 10)
```

Figure 2: Invasion radius

To estimate the spread of the SLF, we extract for each year the radius of the invasion in each sector. We can look at how the radius of the invasion increases over time, when differentiating diffusive spread and jump dispersal.

```
sectors_used = 16

thresholdSectors <- jumpID::attribute_sectors(Thresholds %>%
```

```
                                                  select(year, latitude, longitude, DistToIntro),
                                                nb_sectors = sectors_used,
                                                centroid = c(-75.67534, 40.41524))
```

```
## 2024-08-13 11:03:07.374141 Start sector attribution... Sector attribution completed.
```

```
jumpSectors <- jumpID::attribute_sectors(Jumps %>% select(year, latitude, longitude, DistToIntro),
                                         nb_sectors = sectors_used,
                                         centroid = c(-75.67534, 40.41524))
```

```
## 2024-08-13 11:03:07.386243 Start sector attribution... Sector attribution completed.
```

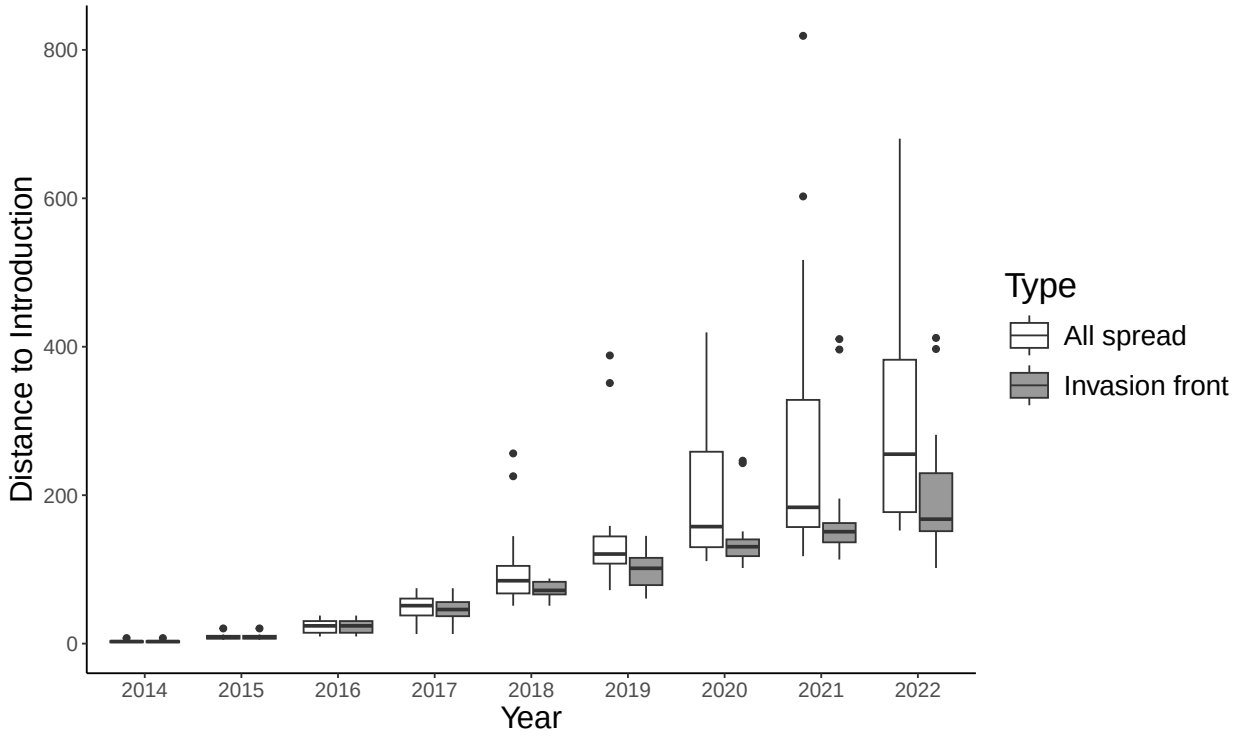
```
thresholdMaxSector <- thresholdSectors %>%
  group_by(year, sectors_nb) %>%
  summarise(maxDistToIntro = max(DistToIntro))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
allMaxSector <- bind_rows(thresholdSectors, jumpSectors) %>%
  group_by(year, sectors_nb) %>%
  summarise(maxDistToIntro = max(DistToIntro))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
radiusData <- bind_rows(allMaxSector %>%
                          mutate(Type = "All spread"),
                        thresholdMaxSector %>% mutate(Type = "Invasion front"))

invasionRadius <- ggplot(stat = "identity") +
  geom_boxplot(data = radiusData, aes(x = as.factor(year), y = maxDistToIntro,
                                      fill = Type)) +
  scale_fill_manual(name = "Type", values = c("All spread" = "white",
                                              "Invasion front" = "gray60")) +
  theme_classic() +
  labs(x = "Year", y = "Distance to Introduction") +
  theme(legend.title = element_text(size = 20),
      legend.text = element_text(size = 16),
      legend.key.size = unit(2, "lines"),
      axis.title = element_text(size = 18),
      axis.text = element_text(size = 12))

invasionRadius
```




```
ggsave(file.path(here::here(), "figures", "2. invasionRadius.jpeg"),
       invasionRadius, height = 6, width = 10)
```

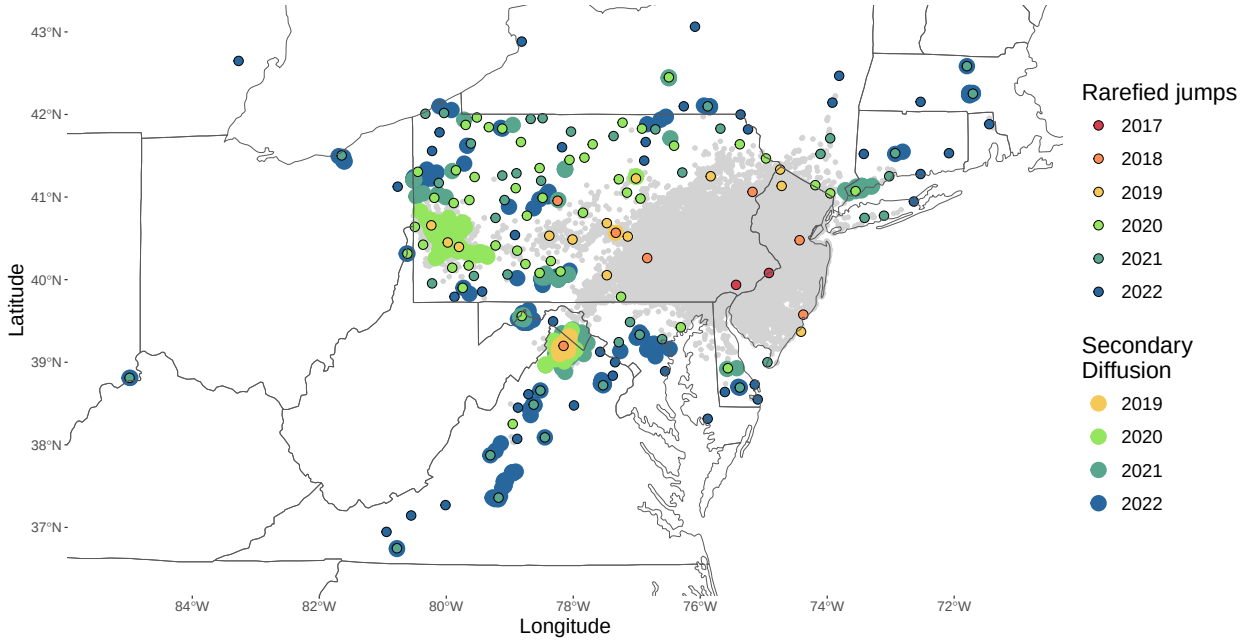
Figure 3: secondary diffusion

3A: Map

Map the points identified as secondary diffusion around dispersal jumps.

```
# Reverse the order of data frames supplied to geom_point
secDiff_map <- ggplot() +
  geom_point(data = grid_data %>% filter(established == TRUE),
             aes(x = longitude, y = latitude),
             col = "lightgrey") +
  geom_point(data = secDiffusion %>% arrange(desc(year)),
             aes(x = longitude, y = latitude,
                 col = as.factor(year)),
             stroke = 2, size = 4) +
  geom_point(data = Jump_clusters %>% arrange(desc(year)),
             aes(x = longitude, y = latitude, col = as.factor(year), fill = as.factor(year)),
             color = "black",
              shape = 21,
             stroke = 0.5, size = 3.5) +
  scale_color_manual(values = c( "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  scale_fill_manual(values = c( "#d53e4f", "#fc8d59", "#f4c957", "#94e65f", "#59a68e", "#27679e")) +
  labs(x = "Longitude", y = "Latitude") +
  theme(legend.position = "right", text = element_text(size = 10),
```

```
        panel.background = element_rect(fill = "white"),
        legend.key = element_rect(fill = "white"),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 16),
        legend.key.size = unit(2, "lines"),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 12)) +
    geom_sf(data = states, alpha = 0) +
    coord_sf(xlim = c(-85.25, -71), ylim = c(36.5, 43)) +
  guides(colour = guide_legend(paste0("Secondary", "\n", "Diffusion")),
         fill = guide_legend("Rarefied jumps"))

secDiff_map
```




```
ggsave(file.path(here::here(), "figures", "3. secDiff_map.jpeg"),
       secDiff_map, height = 8, width = 15)
```

3B: Secondary diffusion Bar Plot

Count how many jumps were followed by secondary diffusion

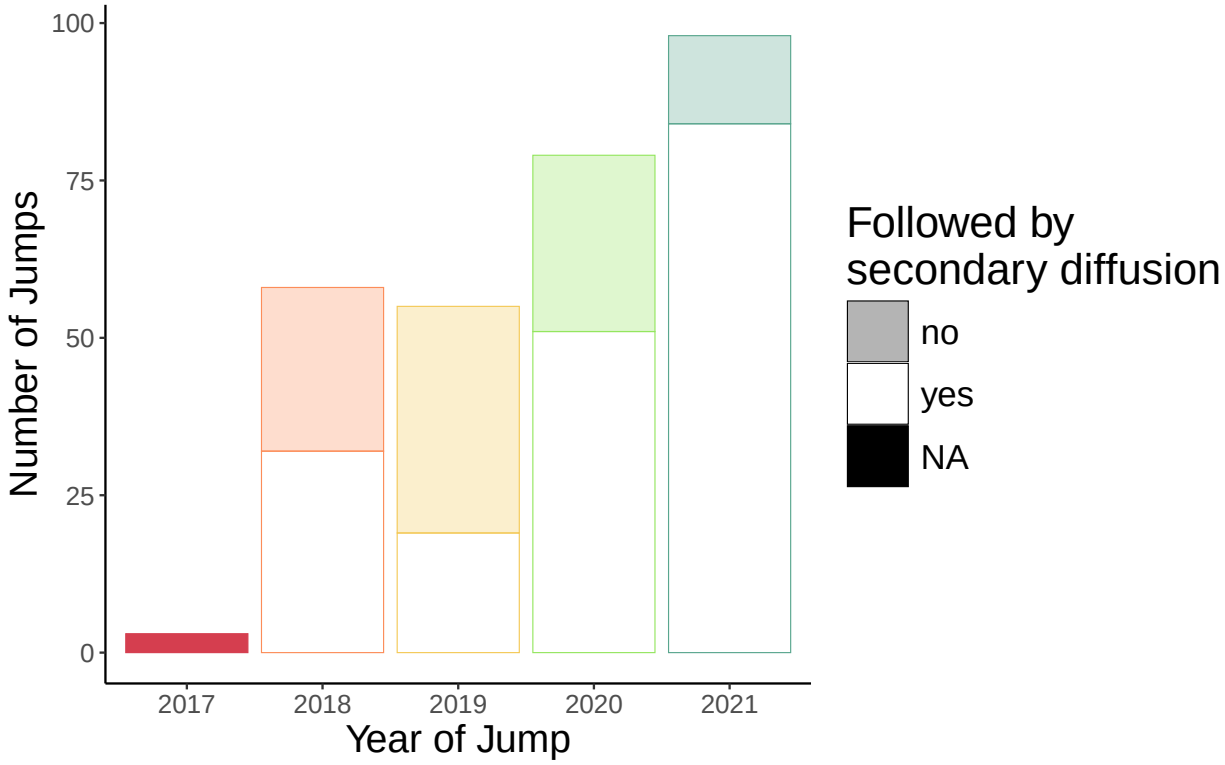
```
# select jumps for which we have data for year n+1
Jumps_2021 <- Jumps %>% filter(year %in% c(2014:2021)) %>%
  mutate(secDiff = NA)
# select diffusion points to check if the jump was already caught up by the
# diffusive spread the year after
diff_secDiff <- setdiff(grid_data %>% filter(established == T),
                        Jumps %>% select(year, latitude, longitude, established, DistToIntro)) # remove
diff <- setdiff(diff_secDiff, secDiffusion) # remove secondary diffusion
```

```r
for (jump in 1:length(Jumps_2021$DistToIntro)){ #for each jump
  # is this jump within 15 km of any diffusion point the year after? ie. caught up by diffusion
  testDiff <- diff %>% filter(year %in% c(min(Jumps_2021$year), Jumps_2021$year[jump]+1))
  pairwise_dist <- geosphere::distGeo(testDiff[,c(3,2)], Jumps_2021[jump,c(4,3)])/1000

  if (min(pairwise_dist) < 15){ Jumps_2021$secDiff[jump] = "caught up"
  } else {
  # is there secondary diffusion within 15 km the year after?
  # calculate pairwise distance with secDiff the year after
    testSecDiff <- secDiffusion %>% filter(year == Jumps_2021$year[jump]+1)
    if (dim(testSecDiff)[1] > 0){
    # check distances
      pairwise_dist <- geosphere::distGeo(testSecDiff[,c(4,3)], Jumps_2021[jump,c(4,3)])/1000
      Jumps_2021$secDiff[jump] = ifelse(min(pairwise_dist) < 15, "yes", "no")
    }
  }
}

Jumps_secDiff <- Jumps_2021 %>%
  group_by(year, secDiff) %>%
  summarise(count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
Jumps_secDiff$secDiff <- factor(Jumps_secDiff$secDiff,
                                levels = c("caught up", "no", "yes"))


#Bar Plot
Jumps_secDiff_plot <- ggplot() +
  geom_bar(data = Jumps_secDiff,
           aes(x = as.factor(year),
               y = count,
               group = secDiff,
               fill = as.factor(year),
               col = as.factor(year),
               alpha = secDiff),
           lwd = .25,
           #position = "dodge"
           stat = "identity") +
  scale_fill_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f",
                               "#59a68e", "#27679e"), guide = "none") +
  scale_color_manual(values = c("#d53e4f", "#fc8d59", "#f4c957", "#94e65f",
                                "#59a68e", "#27679e"), guide = "none") +
  scale_alpha_manual(values = c(0.3, 0, 1)) +
  theme_classic() +
  xlab("Year of Jump") +
  ylab("Number of Jumps") +
  guides(alpha = guide_legend(
    paste0("Followed by", "\n",  "secondary diffusion"),
```

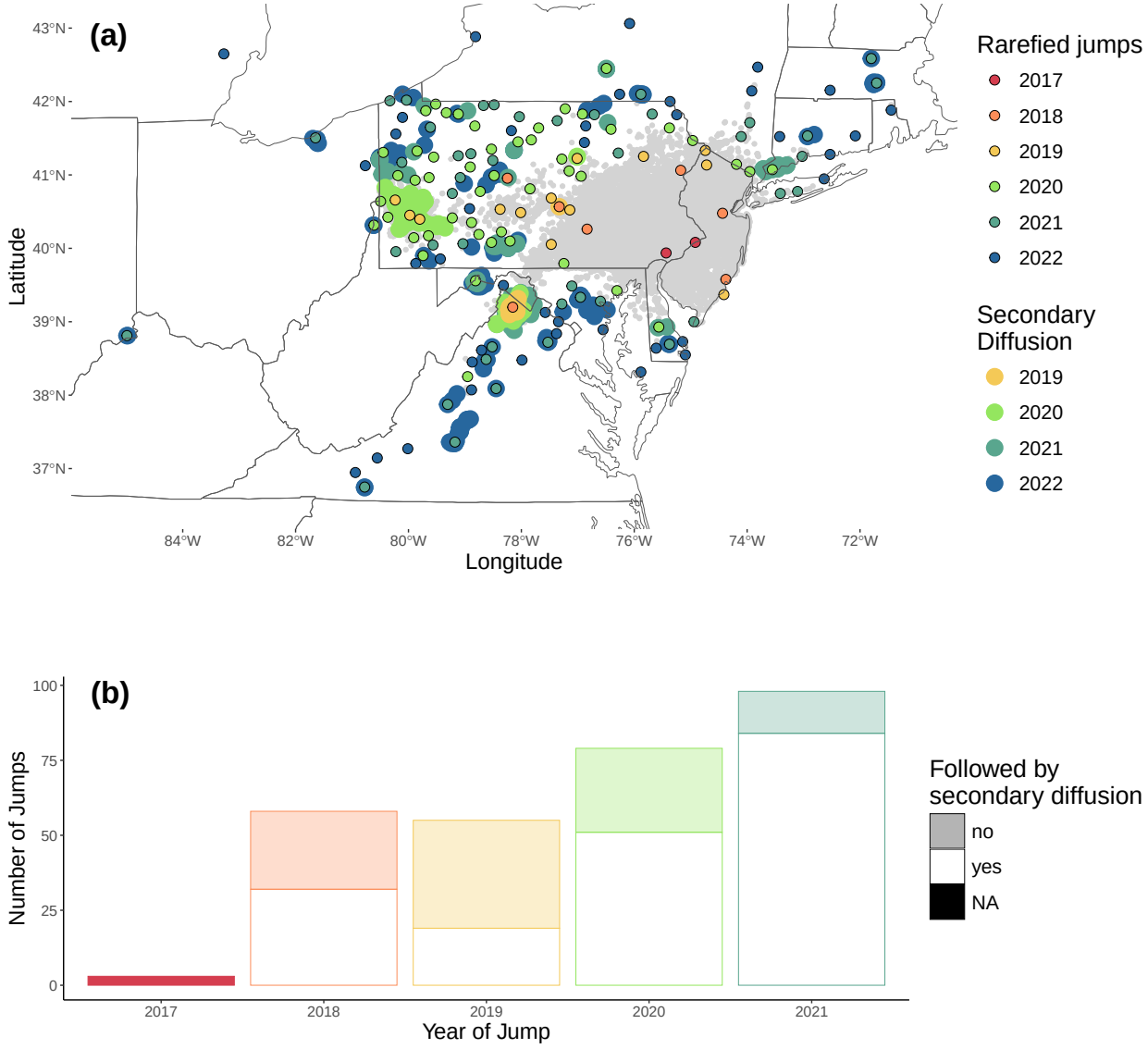
```
    override.aes = list(fill = "black", color = "black", linetype = 1, shape = 21))) +
  theme(legend.title = element_text(size = 20),
        legend.text = element_text(size = 16),
        legend.key.size = unit(2, "lines"),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 12))

Jumps_secDiff_plot
```




```
ggsave(file.path(here::here(), "figures", "3B. secDiff_bar.jpeg"),
       Jumps_secDiff_plot, height = 5, width = 8)
```

Assemble figure 3AB

```
secDiff_graphs_combined <- cowplot::ggdraw() +
  cowplot::draw_plot(secDiff_map, 0, .33, 1, 0.66) +
  cowplot::draw_plot(Jumps_secDiff_plot, 0, 0, 1, 0.33) +
  cowplot::draw_plot_label(c("(a)", "(b)"), c(0.06, 0.06), c(0.90, 0.33), size = 25)

secDiff_graphs_combined
```



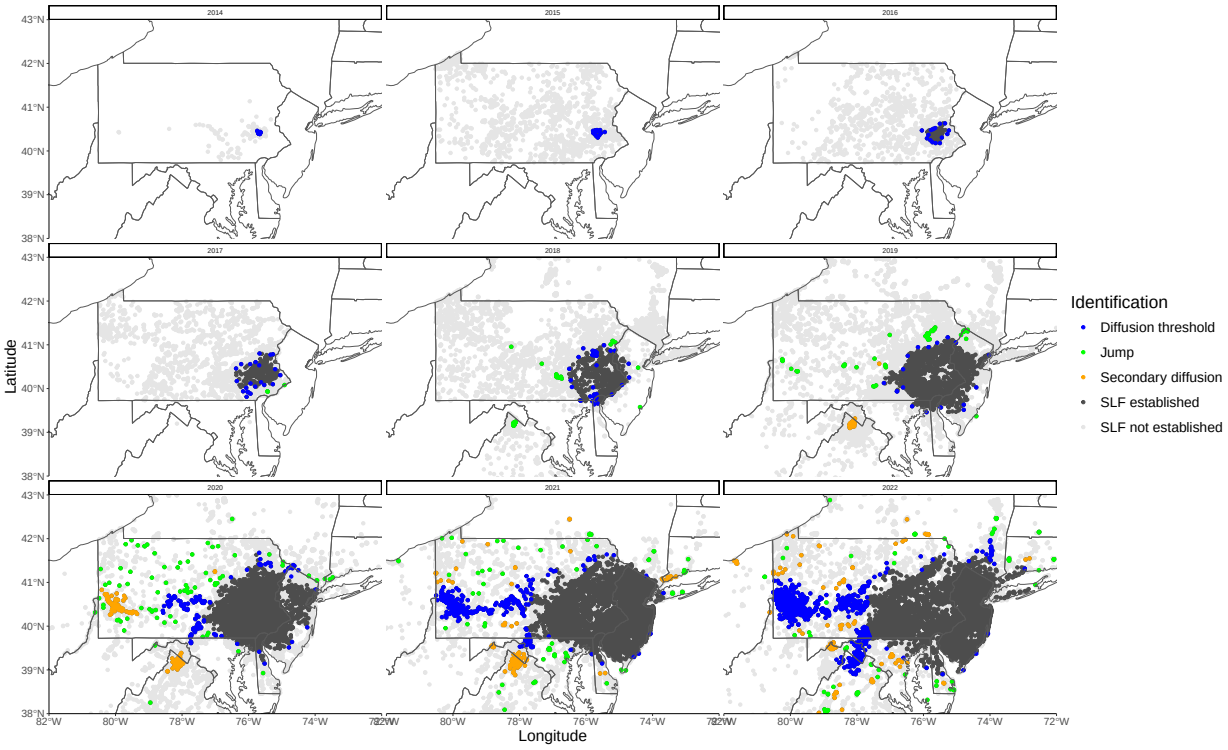

```
ggsave(file.path(here::here(), "figures", "3. secDiff_combined.jpeg"),
       secDiff_graphs_combined, height = 13, width = 13)
```

Figure 4: Visualizing Jumps, Thresholds, and SecDiff

Visualize all results, faceted by year

```
# Make a single object for the map
jumps_wrapper_map <- dplyr::bind_rows(grid_data %>% filter(established == F) %>%
                                        dplyr::mutate(Type = "SLF not established"),
                                      grid_data %>% filter(established == T) %>%
                                        dplyr::mutate(Type = "SLF established"),
```

```
                                    Thresholds %>%
                                      dplyr::mutate(Type = "Diffusion threshold"),
                          Jumps %>% dplyr::mutate(Type = "Jump"),
                          secDiffusion %>% dplyr::mutate(Type = "Secondary diffusion"))

facetedResults <- ggplot(data = states) +
  geom_point(data = jumps_wrapper_map, aes(x = longitude, y = latitude, col = Type)) +
  geom_sf(data = states, alpha = 0) +
  facet_wrap(~year) +
  theme(legend.position = "bottom") +
  theme_classic() +
  scale_color_manual(values = c("blue", "green", "orange", "gray30", "gray90")) +
  xlab("Longitude") + ylab("Latitude") + labs(col = "Identification") +
  coord_sf(xlim = c(-82, -72), ylim = c(38, 43), expand = FALSE) +
  theme(legend.position = "right", text = element_text(size = 10),
      panel.background = element_rect(fill = "white"),
      legend.key = element_rect(fill = "white"),
      legend.title = element_text(size = 20),
      legend.text = element_text(size = 16),
      legend.key.size = unit(2, "lines"),
      axis.title = element_text(size = 18),
      axis.text = element_text(size = 12))

facetedResults
```



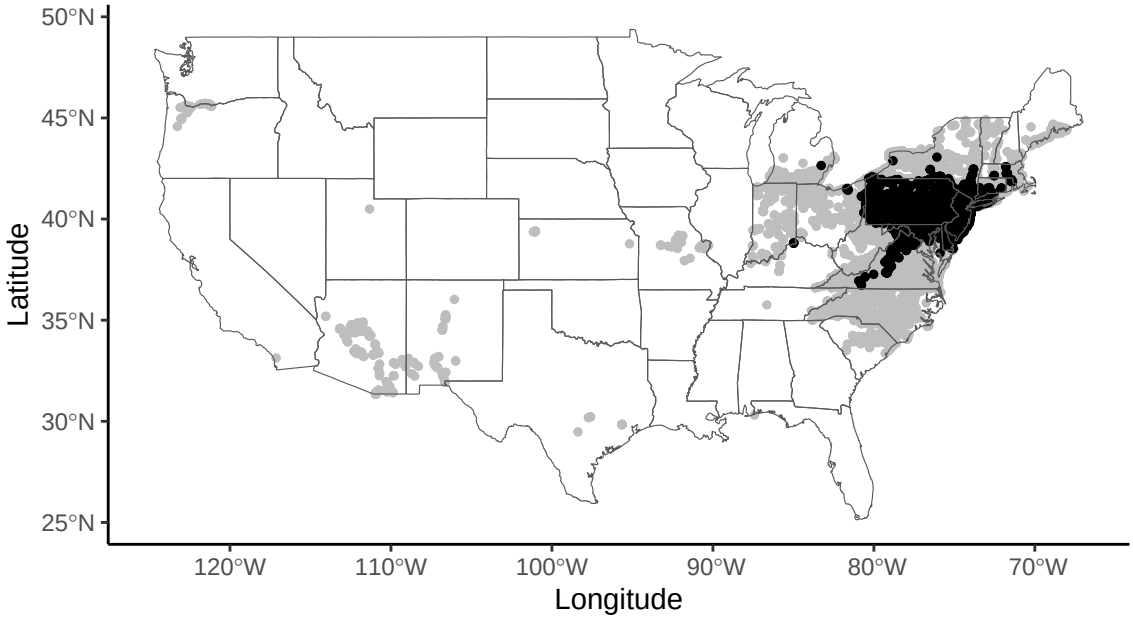

```
ggsave(file.path(here::here(), "figures", "4. faceted_results.jpeg"),
       facetedResults, height = 12, width = 20)
```

Supplementary figures

Figure S1: Map all points

Facet A: Overview of all SLF surveys

```
# plot US map
mapUS <- ggplot(data = states) +
  geom_point(data = grid_data %>% filter(established == FALSE),
             aes(x = longitude, y = latitude), col = "gray", size = 1) +
  geom_point(data = grid_data %>% filter(established == TRUE),
             aes(x = longitude, y = latitude), col = "black", size = 1) +
  geom_sf(alpha = 0) +
  labs(x = "Longitude", y = "Latitude") +
  theme_classic() +
  theme(legend.position = "bottom", legend.key = element_rect(fill = "white", colour = NA))

mapUS
```

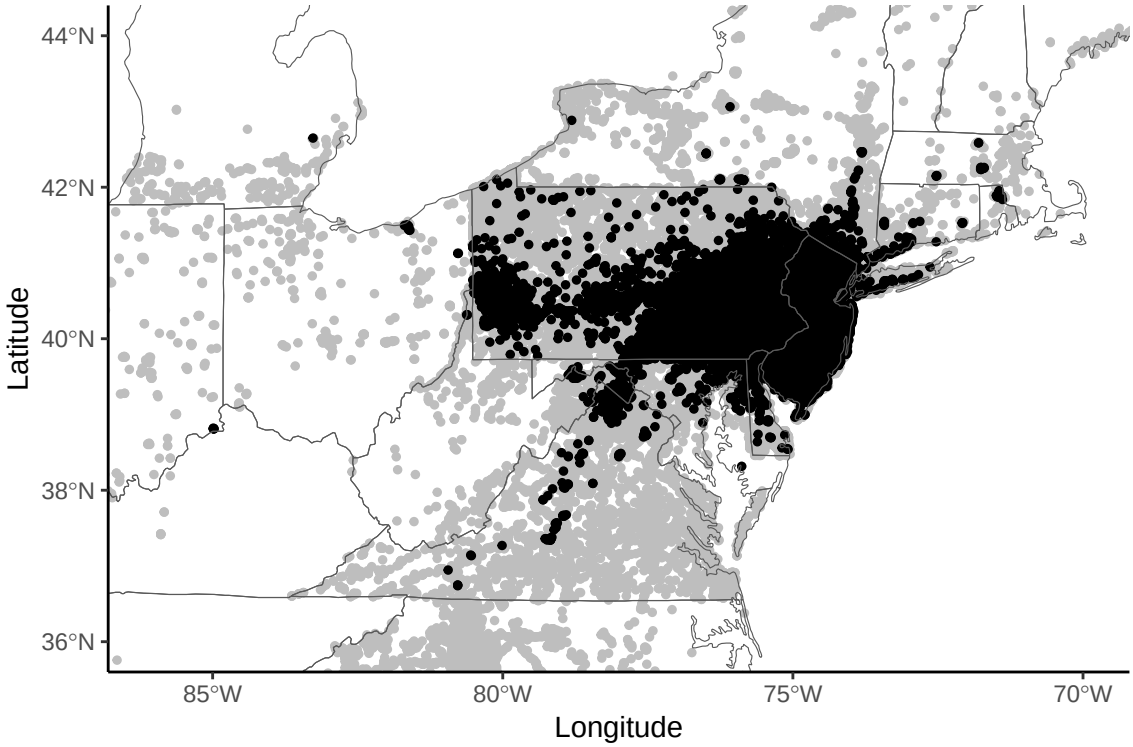



```
# save it
ggsave(file.path(here::here(), "figures", "S1A. points_all.jpg"),
       mapUS, width = 6, height = 6)
```

Facet B: Zoomed map on established SLF

```
# create a variable for a meaningful legend
grid_data %<>% mutate(SLF = ifelse(established == T, "Present", "Absent"))

# plot zoomed map
zoomedMap <- ggplot(data = states) +
  geom_point(data = grid_data %>% filter(established == FALSE),
             aes(x = longitude, y = latitude), col = "gray", size = 1) +
  geom_point(data = grid_data %>% filter(established == TRUE),
             aes(x = longitude, y = latitude), col = "black", size = 1) +
  geom_sf(alpha = 0) +
```

```
  labs(x = "Longitude", y = "Latitude") +
  coord_sf(xlim = c(-86, -70), ylim = c(36, 44)) +
  theme_classic() +
  theme(legend.position="bottom", legend.key = element_rect(fill = "white", colour = NA))

zoomedMap
```

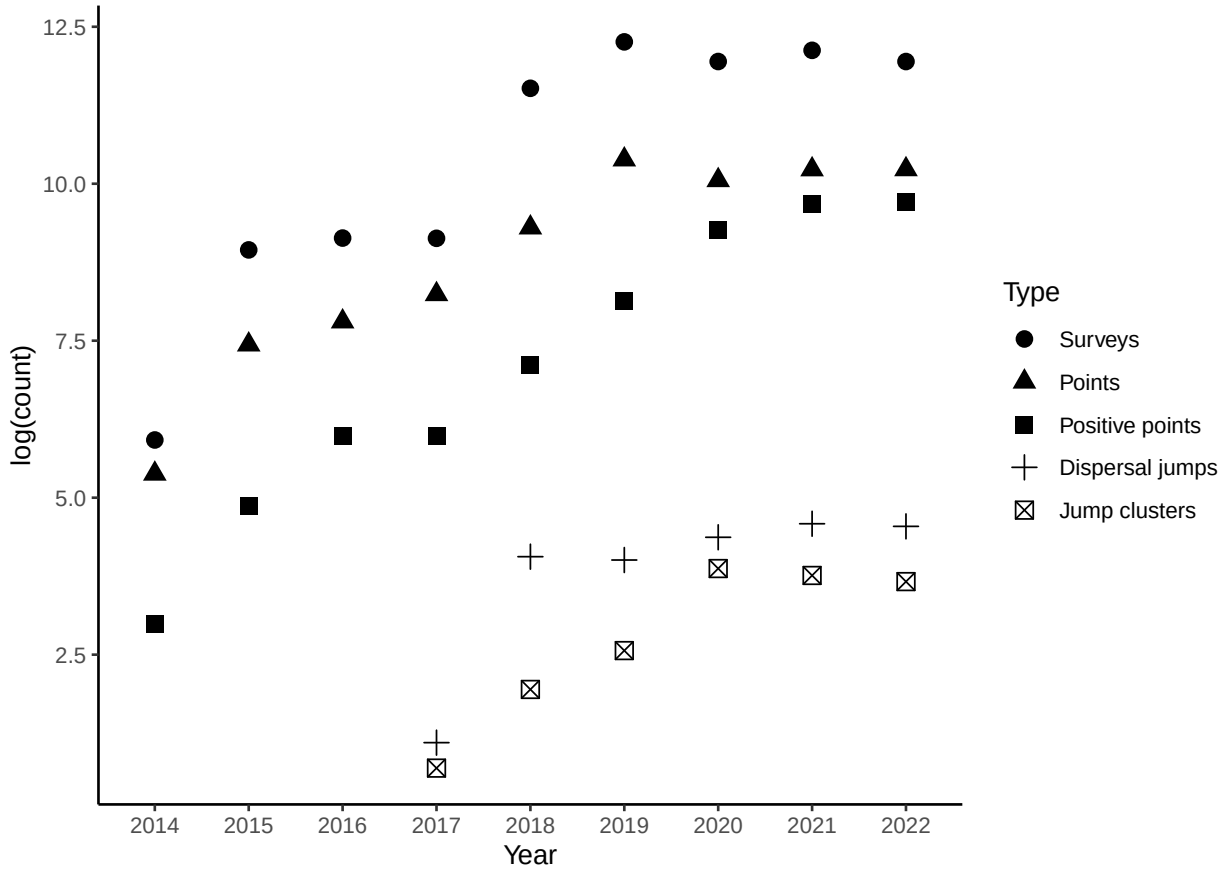



```
# save it
ggsave(file.path(here::here(), "figures", "S1B. zoomed_points.jpg"),
       zoomedMap, width = 6, height = 6)
```

Figure S2: Sampling effort

Show the evolution of the sampling effort and jump occurrences over time

```r
surveys <- as.data.frame(table(slf$bio_year)) %>%
  mutate(Type = "Surveys") %>%
  rename(year = Var1, n = Freq)
points <- as.data.frame(table(grid_data$year)) %>%
  mutate(Type = "Points") %>%
  rename(year = Var1, n = Freq)

positives <- grid_data %>% filter(established == T)
positive_points <- as.data.frame(table(positives$year)) %>%
  mutate(Type = "Positive points") %>%
  rename(year = Var1, n = Freq)

jumps <- Jumps %>%
  count(year) %>%
  mutate(Type = "Dispersal jumps")

clusters <- Jump_clusters %>%
  count(year) %>%
  mutate(Type = "Jump clusters")

effort <- rbind(surveys, points, positive_points, jumps, clusters)
effort$Type <- factor(effort$Type,
                      levels = c("Surveys", "Points", "Positive points",
                                 "Dispersal jumps", "Jump clusters"))

effort_plot <- ggplot() +
  geom_point(data = effort, aes(x = year, y = log(n), shape = Type),
             size = 3) +
  theme_classic() +
  xlab("Year") + ylab("log(count)")

effort_plot
```




```
ggsave(file.path(here::here(), "figures", "S2. number of surveys.jpg"),
       effort_plot, width = 7, height = 5)
```

– end of vignette –