# OdeHeatmap

Nathan G. Earley

2026-01-15

## Intro

I want to look into making heatmaps of Ode records in BC based on the data available on iNat and in Gibson et al (2024) *Keys to the cabinet* pub in Can Ent.

## Methods

### Data

The data were downloaded from the internet and are stored on my own machine. I can send them if you need them.

#### iNaturalist data

For this we'll use the Ode data that I pulled from iNat on Jan 13, 2026.

```r
## Read in the data
iNatOdeDF <- read.csv("data/iNatData/iNatOdesBC2026JAN13.csv",
                      na.strings = c("", " "),
                      stringsAsFactors = FALSE)

## Keep only research grade obs
iNatOdeRG <- iNatOdeDF %>%
  dplyr::filter(quality_grade == "research")

## Keep only CC0, CC-BY, CC-BY-NC records so you don't go to jail for 1B years
iNatOdeRG_CC <- iNatOdeRG %>%
  dplyr::filter(license %in% c("CC0", "CC-BY", "CC-BY-NC"))

## Remove all data with big (>5km) or NA accuracy
iNatOdeRG_CC_acc <- iNatOdeRG_CC %>%
  dplyr::filter(is.na(positional_accuracy) | positional_accuracy < 5000)

## Change the Lat and long columns to Lat and Long
iNatOdeRG_CC_acc <- iNatOdeRG_CC_acc %>%
  dplyr::rename("Longitude" = "longitude") %>%
  dplyr::rename("Latitude" = "latitude")

## Make sure that the iNat data are in the right projection
iNatOde_sf <- iNatOdeRG_CC_acc %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"),
               crs = 4326,
               remove = FALSE)
```

```
## Make a new row called genus_species and remove records that are blank
iNatOde_sf <- iNatOde_sf %>%
  dplyr::mutate(genus_species = ifelse(lengths(strsplit(scientific_name,
                                                        "\\s+")) >= 2,
                                       sub("^([^ ]+ [^ ]+).*$", "\\1",
                                           scientific_name),
                                       NA_character_)) %>%
  dplyr::filter(!is.na(genus_species))
```

### Gibson et al (2024) data

And I'll bring in the digitized Odonata holdings of the BC Entomology Collections from Gibson et al (2024; Can. Entomol. 156(e42): 1–16. doi:10.4039/tce.2024.38).

```
## Read in Gibson data
GibsonDF <- read.csv("data/GibsonData/Gibson2024CanEntSup002.csv",
                     na.strings = c("", " "),
                     stringsAsFactors = FALSE)

## Remove rows with NA Lat Long data
GibsonDF <- GibsonDF %>%
  dplyr::mutate(Longitude = as.numeric(Longitude),
                Latitude  = as.numeric(Latitude)) %>%
  dplyr::filter(!is.na(Longitude), !is.na(Latitude))
```

```
## Warning: There was 1 warning in `dplyr::mutate()`.
## i In argument: `Latitude = as.numeric(Latitude)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
## Restrict the datset to just BC records
GibsonBC <- GibsonDF %>%
  dplyr::filter(Prov_State == "British Columbia")

## Make sure that the Gibson data are in the right projection
GibsonBC_sf <- GibsonBC %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"),
               crs = 4326,
               remove = FALSE)

## Make a new row called genus_species and remove records that are blank
GibsonBC_sf <- GibsonBC_sf %>%
  dplyr::filter(!is.na(Genus), !is.na(Species)) %>%
  dplyr::mutate(genus_species = paste(Genus, Species)) %>%
  dplyr::filter(!is.na(genus_species), genus_species != "", genus_species != " ")

## Fix some old names:
GibsonBC_sf <- GibsonBC_sf %>%
  dplyr::mutate(
    genus_species = dplyr::if_else(
      genus_species == "Cordulegaster dorsalis",
      "Zoraena dorsalis",
      genus_species
    )
  )
```

### Mapping data

Read in a shapefile for BC.

```
bc_sf <- rnaturalearth::ne_states(country = "Canada",
                                  returnclass = "sf") %>%
  dplyr::filter(name == "British Columbia") %>%
  sf::st_transform(3153)

## Vectorize the shp
bc_v <- terra::vect(bc_sf)

## Generate a raster template
r_template <- terra::rast(extent = terra::ext(bc_v),
                          resolution = 25000,   # meters
                          crs = terra::crs(bc_v))
```

## Heatmap generation

As a first step I want to get everything formatted for regular stand alone heat maps.

```
###################
## --- Setup --- ##

iNatOde_sf_proj <- iNatOde_sf %>%
  sf::st_transform(3153)

## Vectorize the points
iNatPts_v <- terra::vect(iNatOde_sf_proj)

###########################
## --- Observations --- ###
iNatObs_r <- terra::rasterize(iNatPts_v,
                              r_template,
                              fun = "count",
                              background = 0)

## Mask to BC
iNatObs_r <- terra::mask(iNatObs_r, bc_v)

######################
## --- Species --- ###
iNatPts_v$species <- iNatOde_sf_proj$genus_species

iNatSpp_r <- terra::rasterize(iNatPts_v,
                               r_template,
                               field = "species",
                               fun = function(x, ...) length(unique(x)),
                               background = 0)

## Mask to BC
iNatSpp_r <- terra::mask(iNatSpp_r, bc_v)
```

```r
########################
## --- Make Maps --- ###

## Convert to dataframes
iNatObs_df <- terra::as.data.frame(iNatObs_r, xy = TRUE, na.rm = FALSE)
iNatSpp_df <- terra::as.data.frame(iNatSpp_r, xy = TRUE, na.rm = FALSE)

names(iNatObs_df)[3]  <- "n_records"
names(iNatSpp_df)[3] <- "species_richness"
```

```r
###################
## --- Setup --- ##

GibsOde_sf_proj <- GibsonBC_sf %>%
  sf::st_transform(3153)

## Vectorize the points
GibsPts_v <- terra::vect(GibsOde_sf_proj)

###########################
## --- Observations --- ###
GibsObs_r <- terra::rasterize(GibsPts_v,
                              r_template,
                              fun = "count",
                              background = 0)

## Mask to BC
GibsObs_r <- terra::mask(GibsObs_r, bc_v)

######################
## --- Species --- ###
GibsPts_v$species <- GibsOde_sf_proj$genus_species

GibsSpp_r <- terra::rasterize(GibsPts_v,
                              r_template,
                              field = "species",
                              fun = function(x, ...) length(unique(x)),
                              background = 0)

## Mask to BC
GibsSpp_r <- terra::mask(GibsSpp_r, bc_v)

########################
## --- Make Maps --- ###

## Convert to dataframes
GibsObs_df <- terra::as.data.frame(GibsObs_r, xy = TRUE, na.rm = FALSE)
GibsSpp_df <- terra::as.data.frame(GibsSpp_r, xy = TRUE, na.rm = FALSE)

names(GibsObs_df)[3]  <- "n_records"
names(GibsSpp_df)[3] <- "species_richness"
```

```r
###################
## --- Setup --- ##
```

```r
Gibs_core <- GibsOde_sf_proj %>%
  dplyr::transmute(
    record_id      = CatalogueNumber,
    source         = "Gibs",
    genus_species  = genus_species,
    event_date     = CollectionDate,
    Latitude       = Latitude,
    Longitude      = Longitude,
    geometry       = geometry
  )

iNat_core <- iNatOde_sf_proj %>%
  dplyr::transmute(
    record_id      = as.character(id),
    source         = "iNat",
    genus_species  = genus_species,
    event_date     = observed_on,
    Latitude       = Latitude,
    Longitude      = Longitude,
    geometry       = geometry
  )

BothOde_sf <- dplyr::bind_rows(Gibs_core, iNat_core)

BothOde_sf_proj <- BothOde_sf %>%
  sf::st_transform(3153)

## Vectorize the points
BothPts_v <- terra::vect(BothOde_sf_proj)

###########################
## --- Observations --- ###
BothObs_r <- terra::rasterize(BothPts_v,
                              r_template,
                              fun = "count",
                              background = 0)

## Mask to BC
BothObs_r <- terra::mask(BothObs_r, bc_v)

######################
## --- Species --- ###
BothPts_v$species <- BothOde_sf_proj$genus_species

BothSpp_r <- terra::rasterize(BothPts_v,
                              r_template,
                              field = "species",
                              fun = function(x, ...) length(unique(x)),
                              background = 0)

## Mask to BC
BothSpp_r <- terra::mask(BothSpp_r, bc_v)
```

```r
########################
## --- Make Maps --- ###

## Convert to dataframes
BothObs_df <- terra::as.data.frame(BothObs_r, xy = TRUE, na.rm = FALSE)
BothSpp_df <- terra::as.data.frame(BothSpp_r, xy = TRUE, na.rm = FALSE)

names(BothObs_df)[3]  <- "n_records"
names(BothSpp_df)[3] <- "species_richness"
```

# Results

## Compare the raw data between the datasets

To start lets see how many records and species are included in each dataset.

```r
nrow(GibsonBC_sf)
```

```
## [1] 27069
```

```r
nrow(iNatOde_sf)
```

```
## [1] 22791
```

```r
nrow(BothOde_sf)
```

```
## [1] 49860
```

```r
length(unique(GibsonBC_sf$genus_species))
```

```
## [1] 93
```

```r
length(unique(iNatOde_sf$genus_species))
```

```
## [1] 87
```

```r
length(unique(BothOde_sf$genus_species))
```

```
## [1] 95
```

```r
## See what's unique to each
gibson_spp <- unique(GibsonBC_sf$genus_species)
inat_spp   <- unique(iNatOde_sf$genus_species)

inat_not_gibson <- setdiff(inat_spp, gibson_spp)
gibson_not_inat <- setdiff(gibson_spp, inat_spp)

inat_not_gibson
```

```
## [1] "Archilestes californicus" "Pantala hymenaea"
```

```r
gibson_not_inat
```

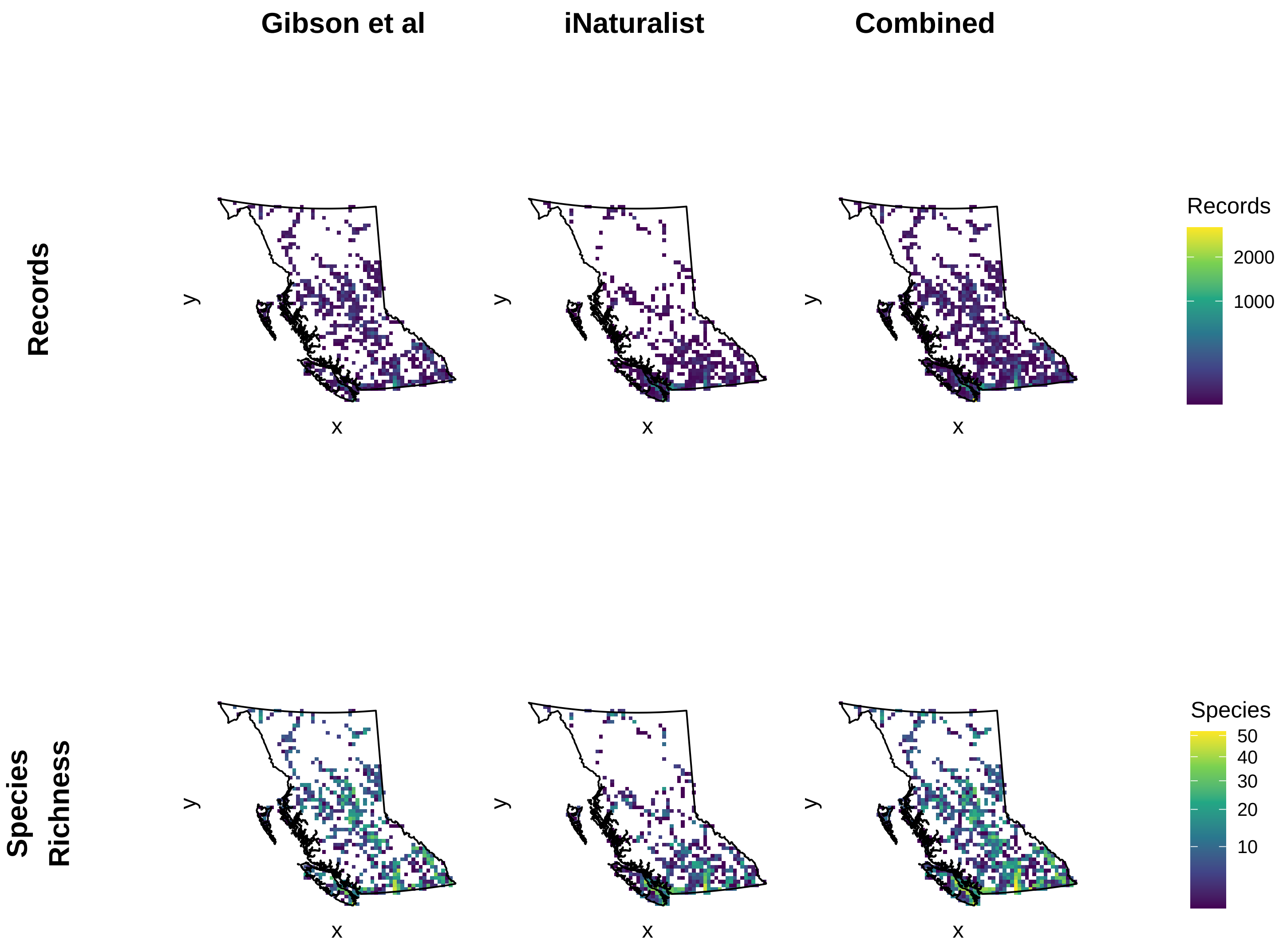
```
## [1] "Sympetrum occidentale"  "Enallagma exsulans"     "Plathemis subornata"
## [4] "Archilestes grandis"    "Hetaerina americana"    "Leucorrhinia frigida"
## [7] "Rhionaeschna cornigera" "Hetaerina vulnerata"
```

Lets see that in a table with the number of records and species available in BC for each dataset.

### Heatmaps

```
# --- Step 1: add a column for plotting, replacing 0 with NA ---
GibsObs_df <- GibsObs_df %>%
  dplyr::mutate(fill_value = ifelse(n_records == 0,
                                    NA_real_,
                                    n_records))

iNatObs_df <- iNatObs_df %>%
  dplyr::mutate(fill_value = ifelse(n_records == 0,
                                    NA_real_,
                                    n_records))

BothObs_df <- BothObs_df %>%
  dplyr::mutate(fill_value = ifelse(n_records == 0,
                                    NA_real_,
                                    n_records))

GibsSpp_df <- GibsSpp_df %>%
  dplyr::mutate(fill_value = ifelse(species_richness == 0,
                                    NA_real_,
                                    species_richness))

iNatSpp_df <- iNatSpp_df %>%
  dplyr::mutate(fill_value = ifelse(species_richness == 0,
                                    NA_real_,
                                    species_richness))

BothSpp_df <- BothSpp_df %>%
  dplyr::mutate(fill_value = ifelse(species_richness == 0,
                                    NA_real_,
                                    species_richness))

# --- Step 2: compute min/max for the scale (optional but ensures same scale across plots) ---
ObsFill_range <- range(c(GibsObs_df$fill_value,
                         iNatObs_df$fill_value,
                         BothObs_df$fill_value),
                       na.rm = TRUE)
SppFill_range <- range(c(GibsSpp_df$fill_value,
                         iNatSpp_df$fill_value,
                         BothSpp_df$fill_value),
                       na.rm = TRUE)

# --- Step 3: make the plots with the same scale ---

## iNat
iNatObsHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = iNatObs_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Records",
                                trans = "sqrt",
                                na.value = "white",
                                limits = ObsFill_range) +
```

```r
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()

iNatSppHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = iNatSpp_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Species",
                                trans = "sqrt",
                                na.value = "white",
                                limits = SppFill_range) +
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()

## Gibson
GibsObsHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = GibsObs_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Records",
                                trans = "sqrt",
                                na.value = "white",
                                limits = ObsFill_range) +
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()

GibsSppHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = GibsSpp_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Species",
                                trans = "sqrt",
                                na.value = "white",
                                limits = SppFill_range) +
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()

## Both
BothObsHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = BothObs_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Records",
                                trans = "sqrt",
                                na.value = "white",
                                limits = ObsFill_range) +
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()

BothSppHeatmap <- ggplot2::ggplot() +
  ggplot2::geom_raster(data = BothSpp_df,
                       ggplot2::aes(x = x, y = y, fill = fill_value)) +
  ggplot2::geom_sf(data = bc_sf, fill = NA, color = "black", linewidth = 0.4) +
  ggplot2::scale_fill_viridis_c(name = "Species",
                                trans = "sqrt",
                                na.value = "white",
                                limits = SppFill_range) +
  ggplot2::coord_sf(crs = sf::st_crs(3153), datum = NA) +
  ggplot2::theme_minimal()




# --- 0: main plots (no legends) ---
p_rec_gibson <- GibsObsHeatmap + guides(fill="none")
p_rec_inat   <- iNatObsHeatmap + guides(fill="none")
p_rec_both   <- BothObsHeatmap + guides(fill="none")
p_spp_gibson <- GibsSppHeatmap + guides(fill="none")
p_spp_inat   <- iNatSppHeatmap + guides(fill="none")
p_spp_both   <- BothSppHeatmap + guides(fill="none")

# --- 1: Generate common legends ---
records_combined <- (GibsObsHeatmap | iNatObsHeatmap | BothObsHeatmap) +
  plot_layout(guides = "collect") & theme(legend.position = "right")

records_legend <- ggdraw(cowplot::get_legend(records_combined))


richness_combined <- (GibsSppHeatmap | iNatSppHeatmap | BothSppHeatmap) +
  plot_layout(guides = "collect") &
  theme(legend.position = "right")

richness_legend <- cowplot::get_legend(richness_combined)

records_legend <- ggdraw(records_legend)
richness_legend <- ggdraw(richness_legend)

# --- 2: Row labels ---
row_records <- ggplot() +
  annotate("text", x=0.5, y=0.5, label="Records", angle=90,
           size=5, fontface="bold") +
  theme_void()
row_richness <- ggplot() +
  annotate("text", x=0.5, y=0.5, label="Species\nRichness", angle=90,
           size=5, fontface="bold") +
  theme_void()

# --- 3: Column titles (add spacer for the left column) ---
col_spacer <- ggplot() + theme_void()  # empty plot for top-left
col_gibson <- ggplot() +
  annotate("text", x=0.5, y=0.5, label="Gibson et al", size=5, fontface="bold") +
  theme_void()
col_inat <- ggplot() +
  annotate("text", x=0.5, y=0.5, label="iNaturalist", size=5, fontface="bold") +
  theme_void()
col_both <- ggplot() +
  annotate("text", x=0.5, y=0.5, label="Combined", size=5, fontface="bold") +
  theme_void()

top_row <- col_spacer | col_gibson | col_inat | col_both | col_spacer # now 5 columns to match bottom r

# --- 4: Bottom row with row labels ---
bottom_rows <-
(row_records | p_rec_gibson | p_rec_inat | p_rec_both | records_legend) /
(row_richness | p_spp_gibson | p_spp_inat | p_spp_both | richness_legend) +
plot_layout(widths = c(0.03, 1, 1, 1), heights = c(1, 1))

# --- 5: Combine column titles with plot grid ---
final_plot <- top_row / bottom_rows + plot_layout(heights = c(0.08, 1))
```


Gibson et al
iNaturalist
Combined
Records
y
x
Records
2000
1000
Species
Richness
Species
50
40
30
20
10