

# Comparison Among Citizen Science Efforts

*Jeff Oliver*

*August 28, 2018*

Comparing estimates of species richness among the two sampling types in the Bioscan project with that of iNaturalist. For these comparisons we are interested in *area* estimates of species richness, rather than estimates for each site.

## Methods

### Setup

Loading dependencies for data wrangling (`tidyr`) and data visualization (`ggplot2`).

```
library("tidyr")
library("ggplot2")
source(file = "bioscan-functions.R")
bioscan <- CompleteBioscan()

# Identify those columns with species data
species.cols <- c(5:33)
```

We are thus going to calculate richness for the area based on Pollard walks, richness for the area based on Malaise traps, and richness for the area based on iNaturalist data. For all these calculations, we will only have a single number; to get an idea of variation, we'll use bootstrap resampling (with replacement) to get a mean richness and some measure of uncertainty about this estimate. For the first two (Pollard walk and Malaise trap), the process will be:

1. Create a bootstrapped sample of all sites for the collection method of interest. For example, if there are 16 rows (sites) of Malaise trap data, we create a new data frame with 16 rows of data sampled with replacement from the original data.
2. Perform richness calculation on that sample.
  1. Extract all columns with species counts
  2. Perform `colSums` calculation on columns with species counts
  3. Count all species with at least one individual present in at least one site
3. Store this richness value as the estimate of richness for that sample.
4. Repeat

For the iNaturalist data, will perform similar approach, but will need only to count number of unique values in species column to determine richness of bootstrapped sample. See `scripts/gbif-processing.sh` and `scripts/gbif-additional-processing.R` for details of the iNaturalist data. We restrict the data in the iNaturalist data set to latitude and longitude boundaries in the bioscan data.

```
# Read in full data
inaturalist <- CleanINaturalist(bioscan.df = bioscan)
```

### Richness calculation

To calculate species richness, we count, for each site/collection method combination, the number of species for which at least one individual was observed. Our data are currently organized so that a single row represents

a single site/collection method combination, so we can perform this operation once for each row and store the data in a new column called `richness`.

```
# Split the two bioscan data into separate data frames for easier calculations
malaise <- bioscan[bioscan$Collection.Method == "Malaise", ]
pollard <- bioscan[bioscan$Collection.Method == "Pollard Walk", ]

malaise.total <- sum(colSums(x = malaise[, species.cols]) > 0)
pollard.total <- sum(colSums(x = pollard[, species.cols]) > 0)
inaturalist.total <- length(unique(inaturalist$species))

# Set up size of bootstrap and data frame to collect results
num.samples <- 100
bootstrapped.df <- data.frame(id = 1:num.samples,
                             Malaise = NA,
                             Pollard = NA,
                             iNaturalist = NA)

for (i in 1:num.samples) {
  bs.malaise <- malaise[sample(x = 1:nrow(malaise), size = nrow(malaise), replace = TRUE), ]
  bootstrapped.df$Malaise[i] <- sum(colSums(x = bs.malaise[, species.cols]) > 0)

  bs.pollard <- pollard[sample(x = 1:nrow(pollard), size = nrow(pollard), replace = TRUE), ]
  bootstrapped.df$Pollard[i] <- sum(colSums(x = bs.pollard[, species.cols]) > 0)

  bs.inaturalist <- inaturalist[sample(x = 1:nrow(inaturalist), size = nrow(inaturalist), replace = TRUE), ]
  bootstrapped.df$iNaturalist[i] <- length(unique(bs.inaturalist$species))
}

# Calculate means so we can report those
mean.bs.pollard <- mean(bootstrapped.df$Pollard)
mean.bs.malaise <- mean(bootstrapped.df$Malaise)
mean.bs.inaturalist <- mean(bootstrapped.df$iNaturalist)
```

## Results

Looking at totals of species in the three data sources:

|                       | Malaise | Pollard | iNaturalist |
|-----------------------|---------|---------|-------------|
| Observed richness     | 18      | 27      | 23          |
| Bootstrapped estimate | 16.36   | 23.99   | 20.7        |

We can compare the three sources of richness visually with a boxplot (after reshaping our data):

```
bootstrap.long <- gather(data = bootstrapped.df, key = "Collection.Method", value = "Richness", -id)

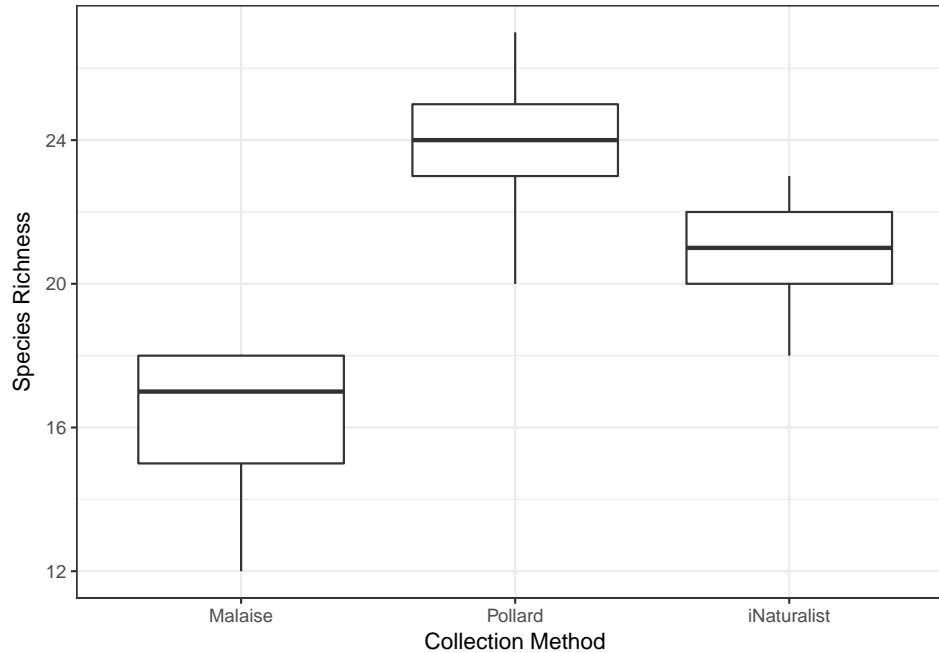
# Re-leveling the Collection.Method so it displays in preferred order
bootstrap.long$Collection.Method <- factor(bootstrap.long$Collection.Method,
                                           levels = c("Malaise", "Pollard", "iNaturalist"))

richness.boxplot <- ggplot(data = bootstrap.long, mapping = aes(x = Collection.Method, y = Richness)) +
  geom_boxplot() +
```

```

xlab(label = "Collection Method") +
ylab(label = "Species Richness") +
theme_bw()
print(richness.boxplot)

```



```

ggsave(filename = "output/figure-bootstrap-richness.png", plot = richness.boxplot)

```

## Saving 6.5 x 4.5 in image

We also would like to glance at the overlap and differences among the three collections, a Venn diagram works best. **Bleah** This is pretty ugly and should be replaced with a **ggplot2** approach. See example at <https://scriptsandstatistics.wordpress.com/2018/04/26/how-to-plot-venn-diagrams-using-r-ggplot2-and-ggforce/>.

```

# Need to count the 7 categories
# Malaise only
# Pollard only
# iNaturalist only
# Malaise + Pollard
# Malaise + iNaturalist
# Pollard + iNaturalist
# Malaise + Pollard + iNaturalist

# Start by pulling out species names for the two bioscans (only those with non-
# zero observations)
malaise.totals <- colSums(x = malaise[, species.cols])
malaise.species <- names(x = malaise.totals)[malaise.totals > 0]

pollard.totals <- colSums(x = pollard[, species.cols])
pollard.species <- names(x = pollard.totals)[pollard.totals > 0]

inaturalist.species <- unique(as.character(inaturalist$species))

all.species <- union(x = malaise.species,

```

```

        y = union(x = pollard.species, y = inaturalist.species))

species.in.methods <- data.frame(species = all.species,
                                malaise = all.species %in% malaise.species,
                                pollard = all.species %in% pollard.species,
                                inaturalist = all.species %in% inaturalist.species)
species.in.methods <- species.in.methods[order(species.in.methods$species), ]

# OMG this is so freaking ugly

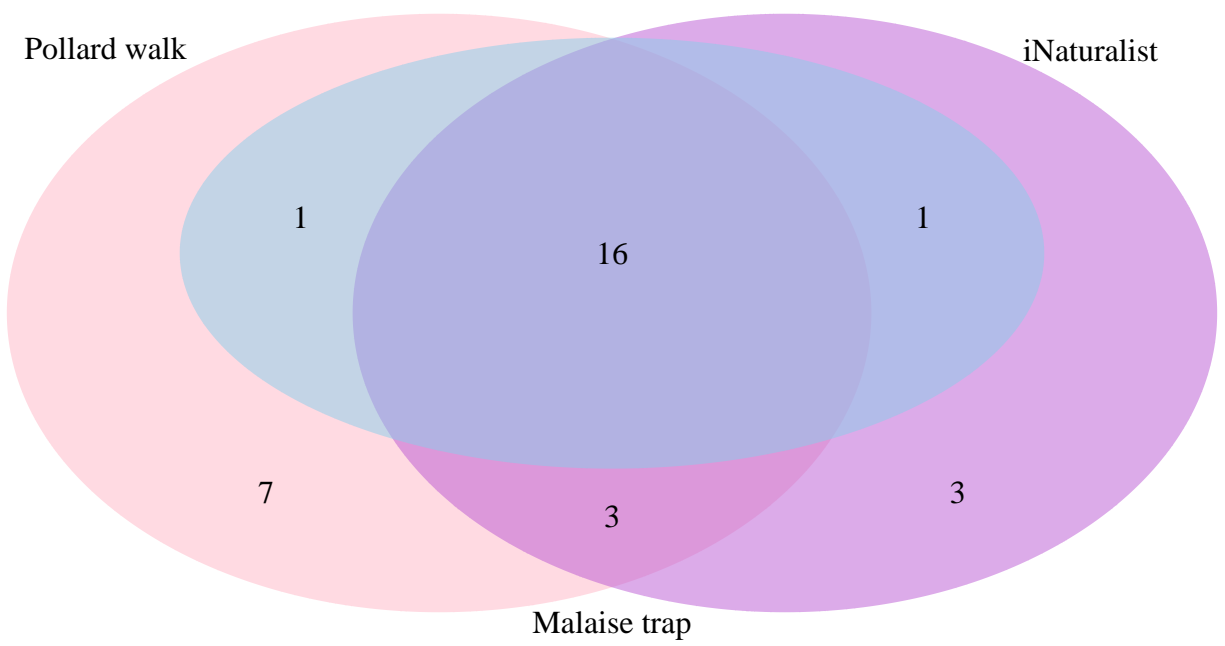
malaise.only <- sum(species.in.methods$malaise &
                   !species.in.methods$pollard &
                   !species.in.methods$inaturalist)
pollard.only <- sum(species.in.methods$pollard &
                   !species.in.methods$malaise &
                   !species.in.methods$inaturalist)
inaturalist.only <- sum(species.in.methods$inaturalist &
                       !species.in.methods$malaise &
                       !species.in.methods$pollard)
malaise.pollard <- sum(species.in.methods$malaise &
                      species.in.methods$pollard)
malaise.inaturalist <- sum(species.in.methods$malaise &
                          species.in.methods$inaturalist)
pollard.inaturalist <- sum(species.in.methods$pollard &
                          species.in.methods$inaturalist)
all.three <- sum(species.in.methods$malaise &
                species.in.methods$pollard &
                species.in.methods$inaturalist)
# install.packages("VennDiagram")
library("VennDiagram")

## Loading required package: grid

## Loading required package: futile.logger

library("grid")
grid.newpage()
draw.triple.venn(area1 = length(malaise.species),
                 area2 = length(pollard.species),
                 area3 = length(inaturalist.species),
                 n12 = malaise.pollard,
                 n13 = malaise.inaturalist,
                 n23 = pollard.inaturalist,
                 n123 = all.three,
                 category = c("Malaise trap", "Pollard walk", "iNaturalist"),
                 lty = "blank",
                 fill = c("skyblue", "pink1", "mediumorchid"))

```



```
## (polygon[GRID.polygon.483], polygon[GRID.polygon.484], polygon[GRID.polygon.485], polygon[GRID.polygon.486])
```