Data visualization using ecocomDP

Preparing and loading data

```
library(tidyverse)
library(neonUtilities)
devtools::load_all("C:/Users/savan/Documents/GitHub/savannahrayegonzales/ecocomDP")
#library(ecocomDP)
```

The ecocomDP retrieves datasets from NEON, EDI, and other repositories. This search function allows you to input a keyword and search for dataset that match your needs. In the example below, the keyword "invertebrates" is used. Uncommenting the second line will allow you to view the table returned from the search function.

```
search_result <- ecocomDP::search_data(text = "invertebrates")
#View(search_result)
```

In this example, we are going to use NEON's macroinvertebrate dataset, the 7th and last row in the search result table. To pull this data, we use ecocomDP's read_data function, and input the dataset's id as the first argument. Since this data is from the NEON repository, some optional arguments include a list of sites, start and end dates for the data, and a user-specific API token. More information on using a NEON token can be found here: https://www.neonscience.org/resources/learning-hub/tutorials/neon-api-tokens-tutorial

```
inv <- ecocomDP::read_data(
  id = "neon.ecocomdp.20120.001.001",
  site = c('ARIK','CARI','MAYF'),
  startdate = "2017-06",
  enddate = "2020-03",
  token = NEON_TOKEN, #this line should be commented or removed if not using a NEON token
  check.size = FALSE)
```

```
## Finding available files
##   |                                                                               |
##
## Downloading files totaling approximately 2.178112 MB
## Downloading 29 files
##   |                                                                               |
##
## Unpacking zip files using 1 cores.
## Stacking operation across a single core.
## Stacking table inv_fieldData
## Stacking table inv_persample
## Stacking table inv_pervial
## Stacking table inv_taxonomyProcessed
## Stacking table inv_taxonomyRaw
## Copied the most recent publication of validation file to /stackedFiles
## Copied the most recent publication of categoricalCodes file to /stackedFiles
## Copied the most recent publication of variable definition file to /stackedFiles
## Finished: Stacked 5 data tables and 3 metadata tables!
## Stacking took 0.78795 secs
```

Now the NEON macroinvertebrate data collected at sites ARIK, CARI, and MAYF from June 2017 to March 2020 is stored in the global environment. We can now begin analyzing and plotting this data using the ecocomDP package.

Data model

First, it's important to understand how ecocomDP data model works. There are a total of eight tables. The main three tables are titled observation, location, and taxon. The remaining five tables provide additional information about the dataset, but are not used in the following plotting functions. The image below depicts each table and some of the variables they contain.

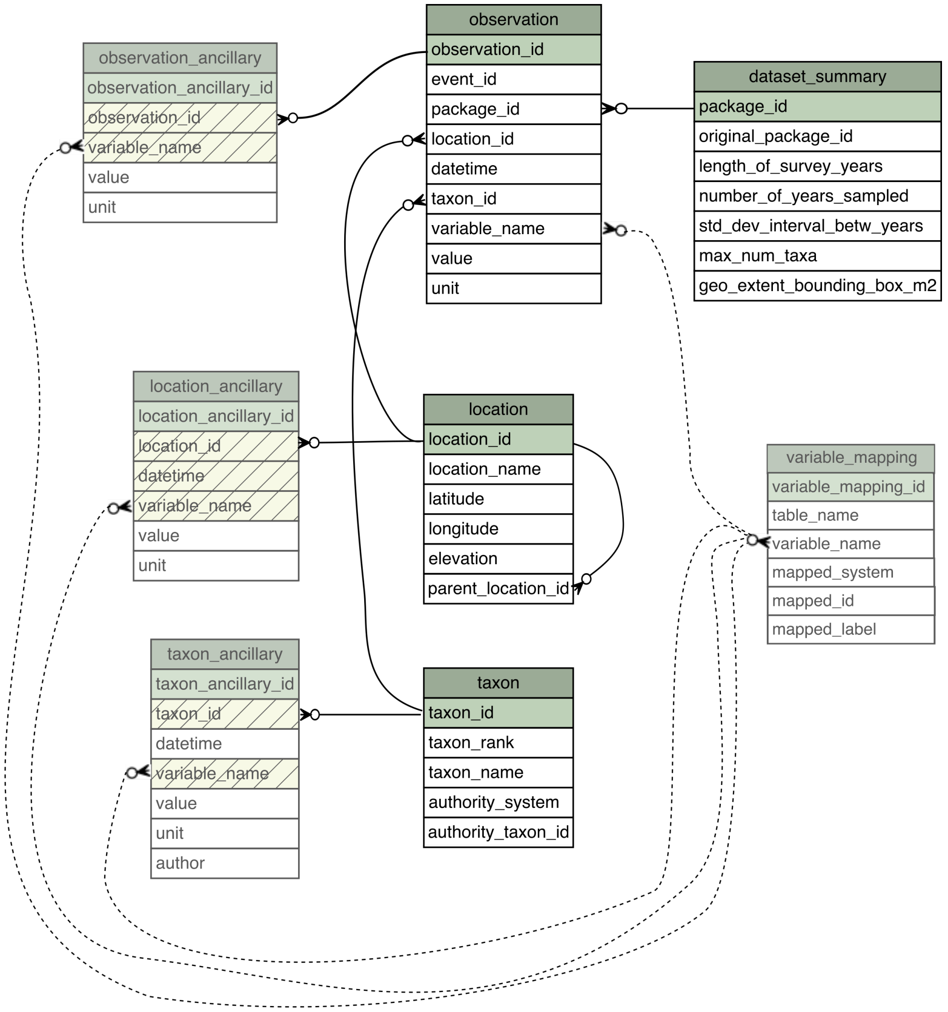

Figure 1: Image of the eight tables stored in the ecocomDP data model

Plotting functions

To begin, you may be interested in viewing where the sites are located. The plot_sites function shows each sites' location over a map of the US, based on their coordinates stored in the location table. This function is unique in that its first argument is a flattened data table. To get this flattened table, we use ecocomDP's flatten_data function.

```
inv_flat <- ecocomDP::flatten_data(inv[[1]]$tables)
ecocomDP::plot_sites(inv_flat,
                     inv[[1]]$tables$observation$location_id)
```

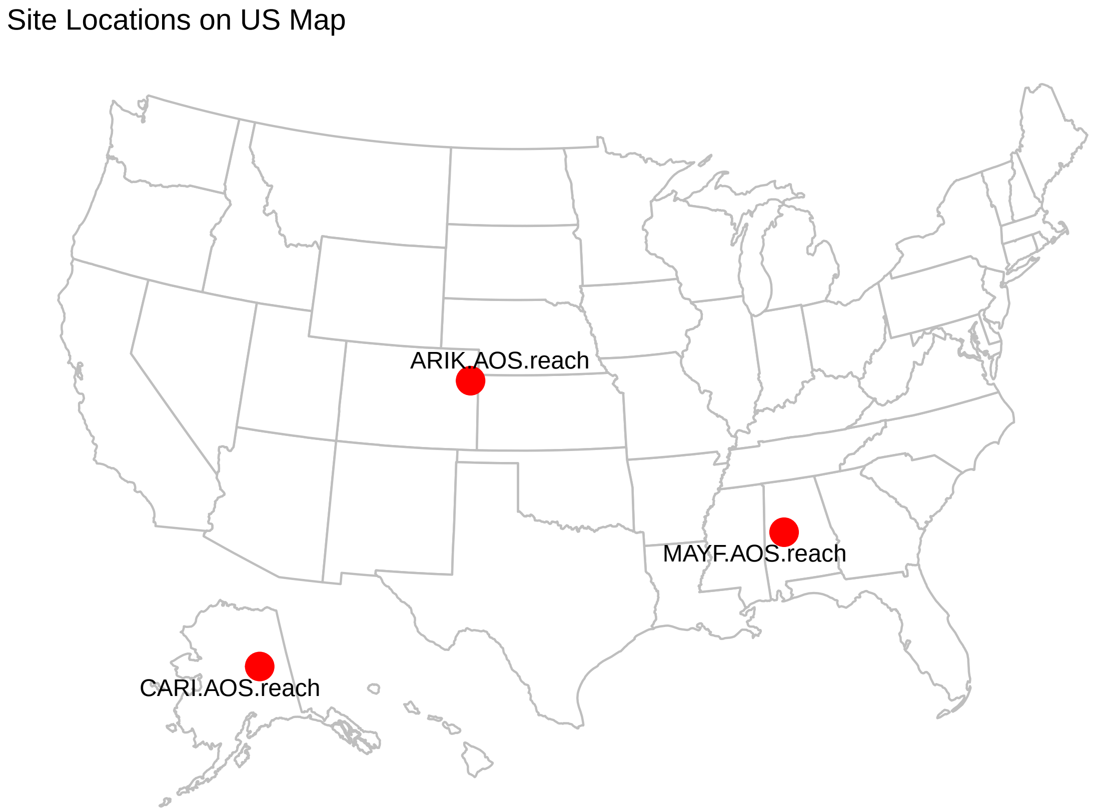

Figure 2: Map of US with location of each site in the dataset.

A quick internet search can confirm that each of the NEON sites are plotted in the correct location. ARIK refers to the Arikaree River that runs primarily through the state of Colorado, CARI refers to the Caribou Creek in central Alaska, and MAYF refers to the Mayfield Creek in west-central Alabama.

Since we are analyzing macroinvertebrate data, we may be interested in viewing which taxa ranks are most commonly recorded. To view this information, we use the plot_taxa_rank function. This function takes both the observation table and the taxon table as its first two arguments and produces a bar graph displaying the frequencies of each taxon rank.

```
ecocomDP::plot_taxa_rank(inv[[1]]$tables$observation,
                         inv[[1]]$tables$taxon,
                         inv[[1]]$tables$observation$location_id)
```

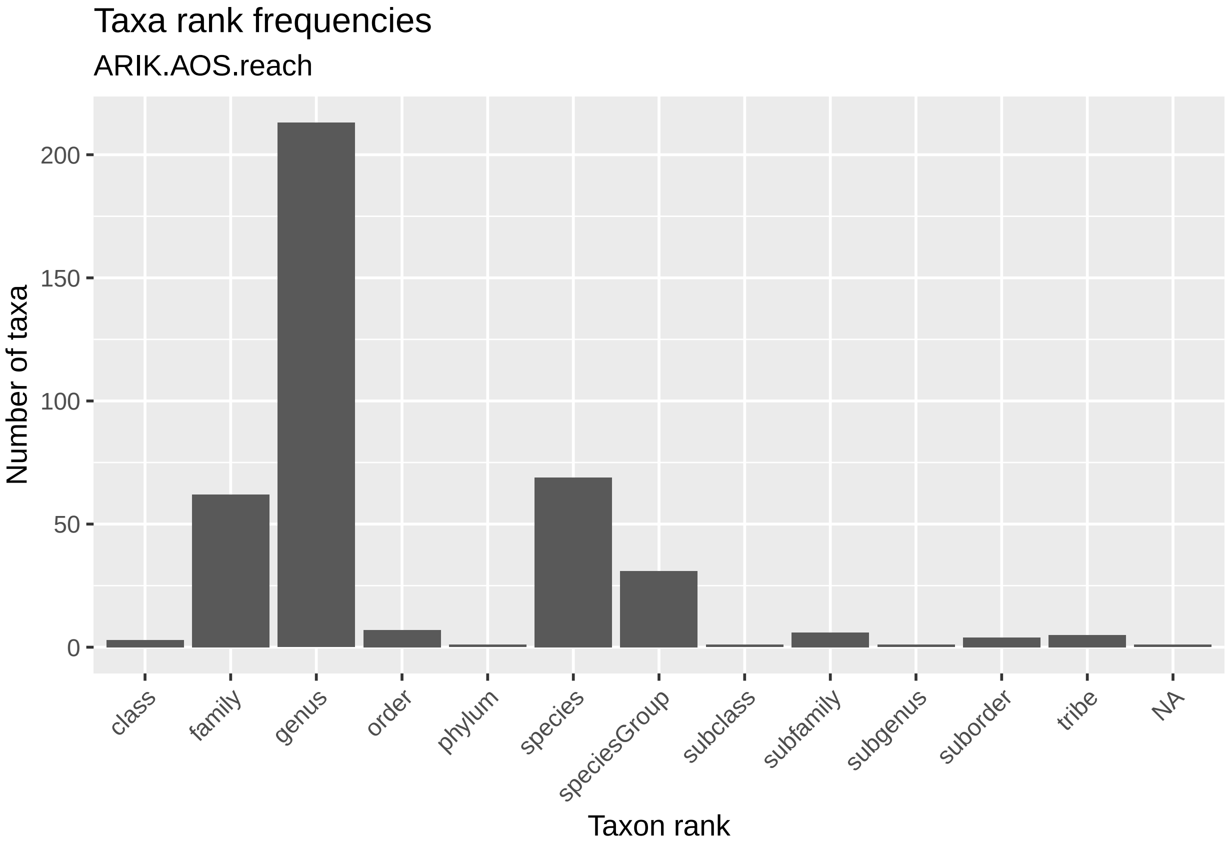

Figure 3: Over all three sites, the taxon rank genus is most commonly recorded.

Similarly, plot_taxa_rank_by_site plots the frequencies of each rank, but divides the graph into each site. In this example, there are three sites, whose names appear at the top of each bar graph.

```
ecocomDP::plot_taxa_rank_by_site(inv[[1]]$tables,
                                 inv[[1]]$tables$observation$location_id)
```

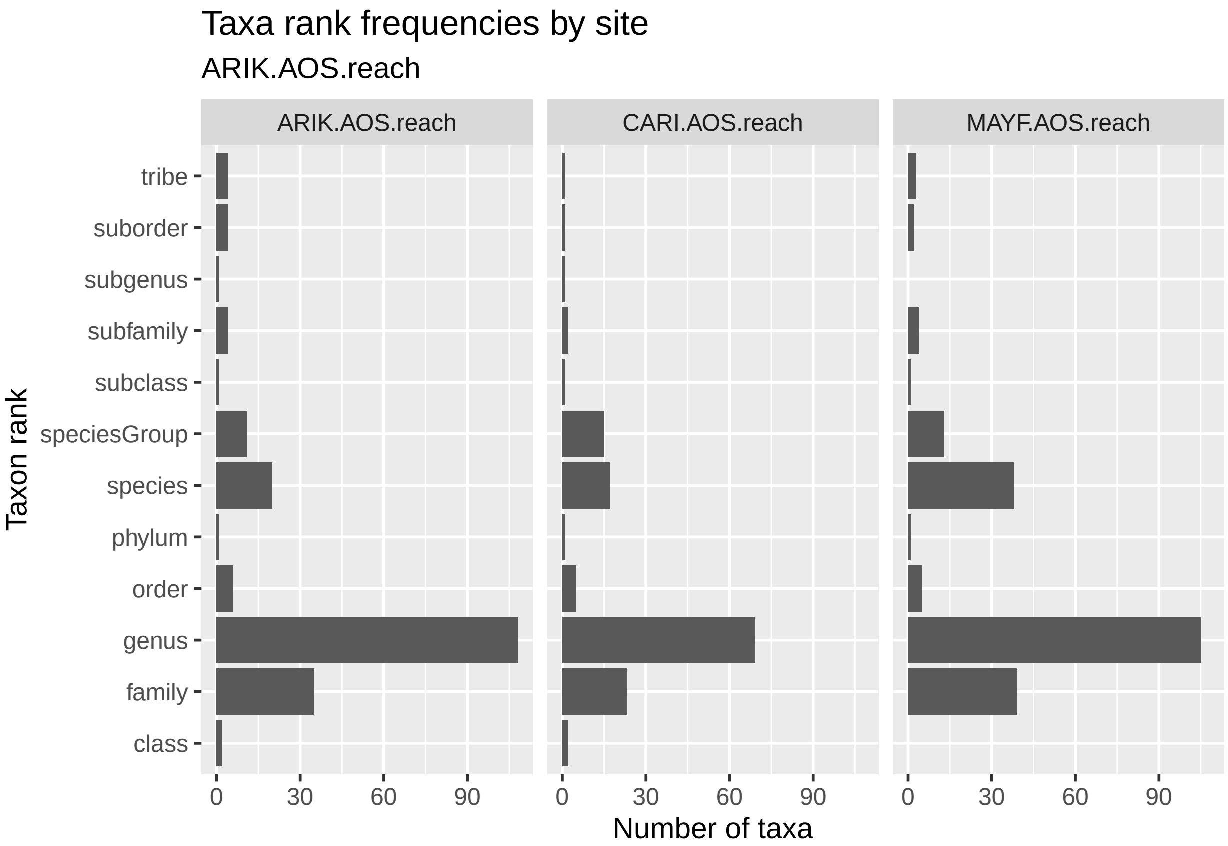

Figure 4: The taxon rank genus is most commonly recorded at all three sites individually.

The plot_stacked_taxa_by_site function plots the frequencies of each taxa gathered from both sites. Each of the sites are color coded; in this example, red represents ARIK and green represents CARI, and blue represents MAYF. The argument rank="order" is added so the taxon orders are plotted. It is important to note this does not include more specific ranks (ex: does not include the order when the observation's taxon_rank=species, despite that species belonging to the same order).

```
ecocomDP::plot_stacked_taxa_by_site(inv[[1]]$tables,
                                    inv[[1]]$tables$observation$location_id,
                                    rank="order")
```

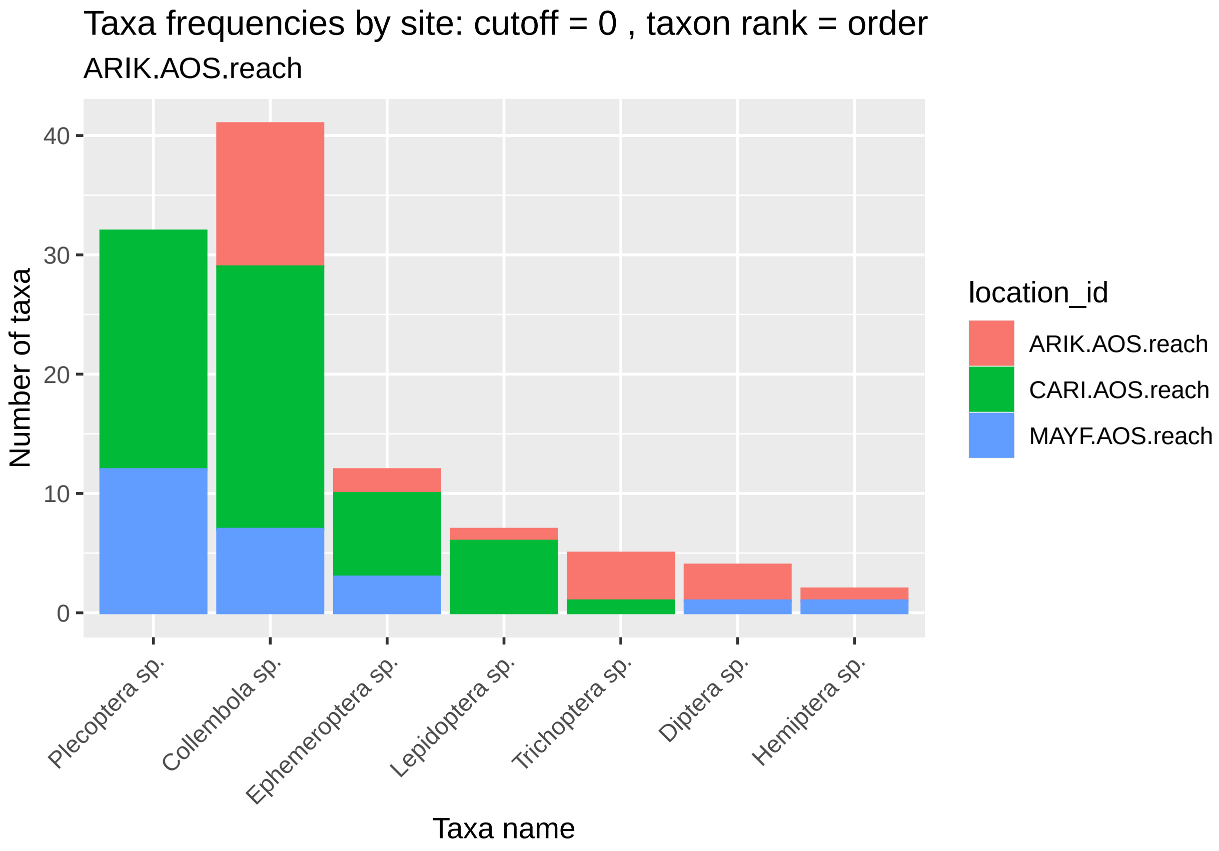

Figure 5: Collembola sp. is the most commonly recorded order at over all three sites.

When the rank is not specified, all taxa observed are plotted. Since some datasets may have a large number of taxa, users may specify a cutoff such that only taxa whose counts are greater than the cutoff are shown. This example sets the cutoff equal to 30, so any taxa with occurrences less than 30 are excluded. Again, this plot does not account for more specific ranks. This can be corrected in the future by implementing a taxon hierarchy that matches any specific rank to its higher ranks.

```
ecocomDP::plot_stacked_taxa_by_site(inv[[1]]$tables,
                                    inv[[1]]$tables$observation$location_id,
                                    cutoff=30)
```

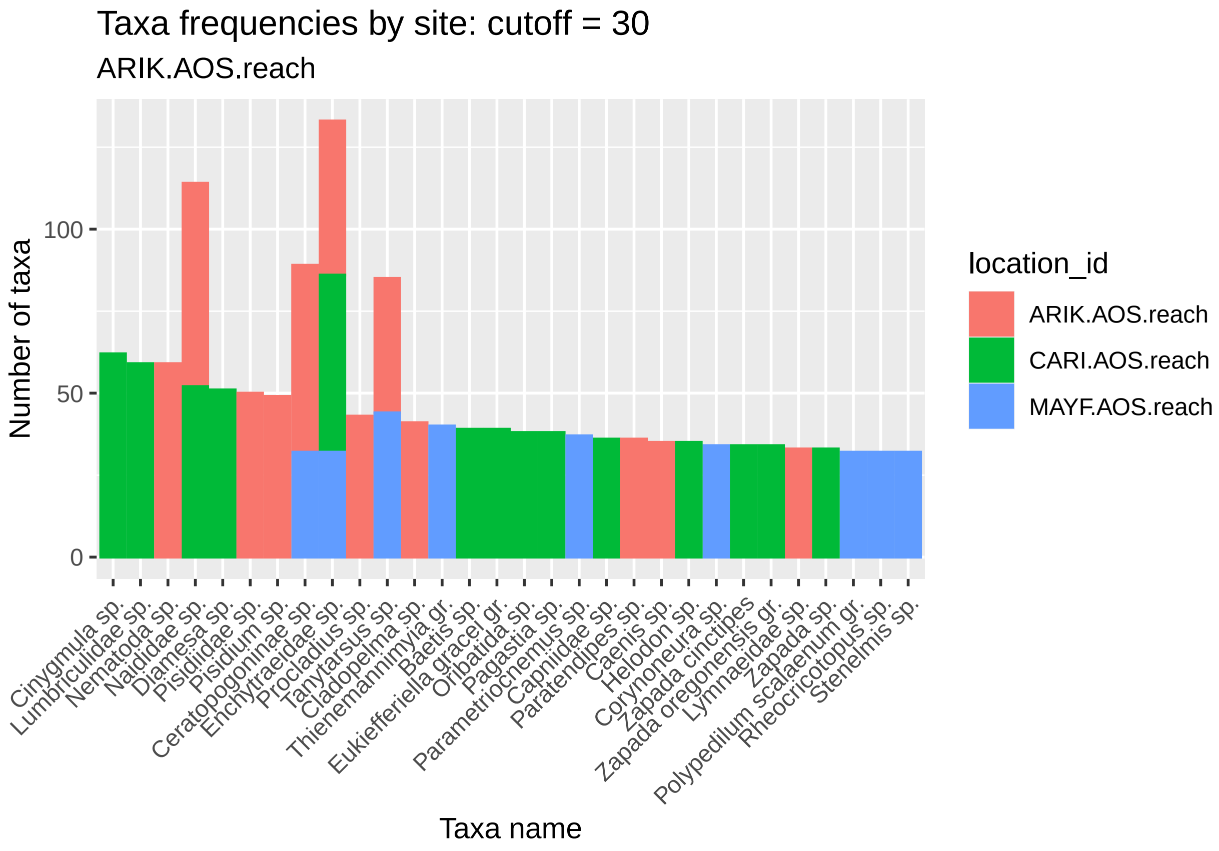

Figure 6: Most commonly recorded taxa at each site. Many of these taxa with high occurences are only present at one of the three sites.

To show the averages of each taxa, we use the plot_faceted_densities function. This plot is split by sites and only includes the taxa from a specified rank; in this example, we are observing the averages of each order. Similar to the plots above, this does not include more specific ranks.

```
ecocomDP::plot_faceted_densities(inv[[1]]$tables,
                                 inv[[1]]$tables$observation$location_id,
                                 rank="order")
```

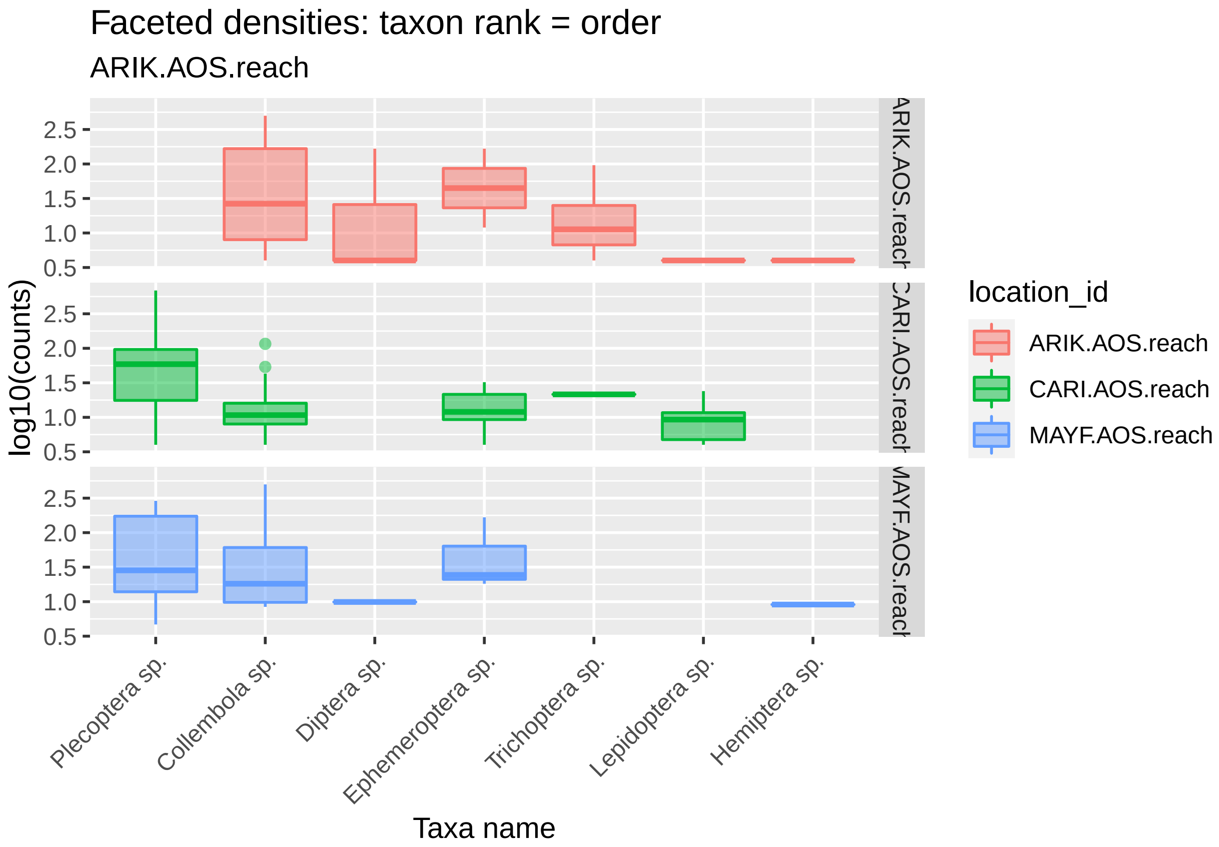

Figure 7: Averages of each order at individual sites.

Existing plotting functions in the ecocomDP package

The function plot_taxa_diversity records the number of unique taxa observed at each site over time.

```
ecocomDP::plot_taxa_diversity(inv[[1]]$tables$observation,
                              inv[[1]]$tables$observation$taxon_id)
```

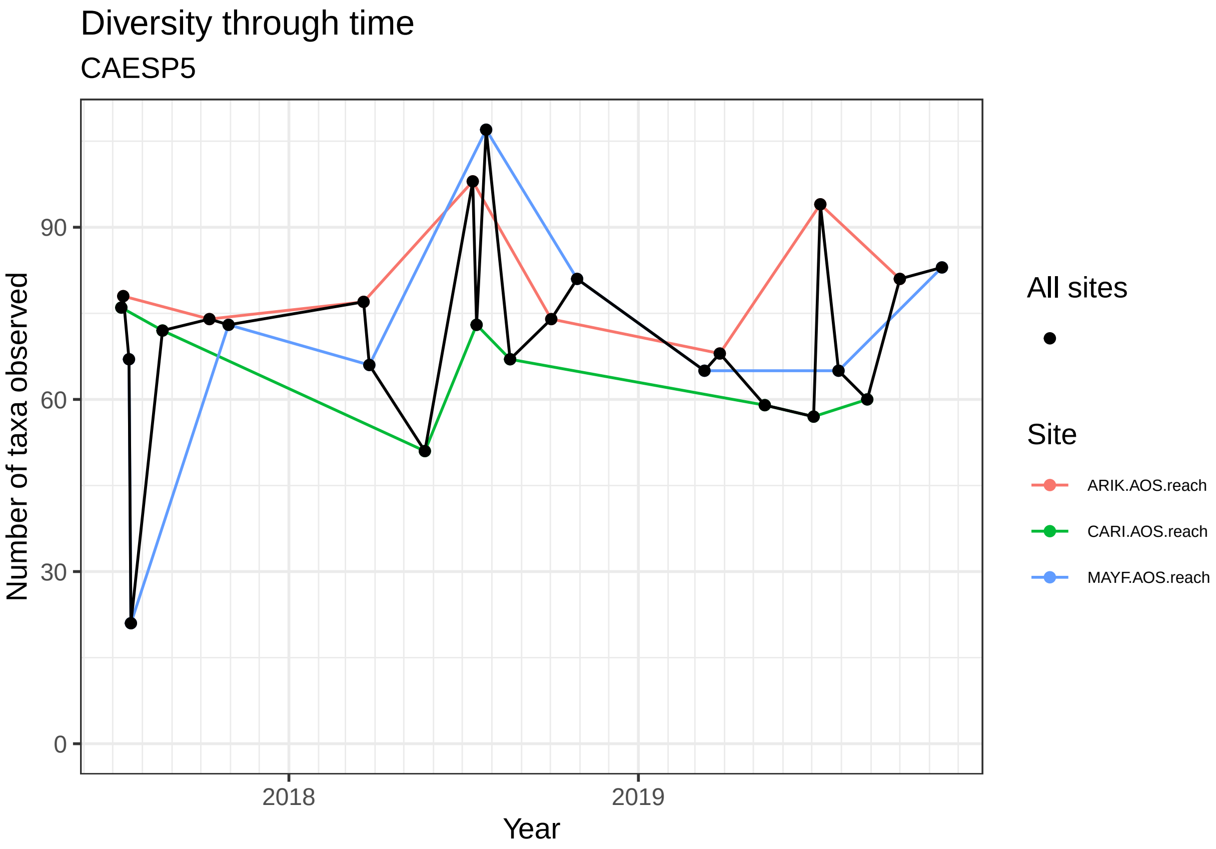

Figure 8: Number of taxa across all three sites over time.

To visualize when samples were taken, the plot_taxa_sample_time function is used. This function allows you to compare sampling times across all three sites, as each month is represented by a vertical line.

```
ecocomDP::plot_taxa_sample_time(inv[[1]]$tables$observation,
                                inv[[1]]$tables$observation$taxon_id)
```

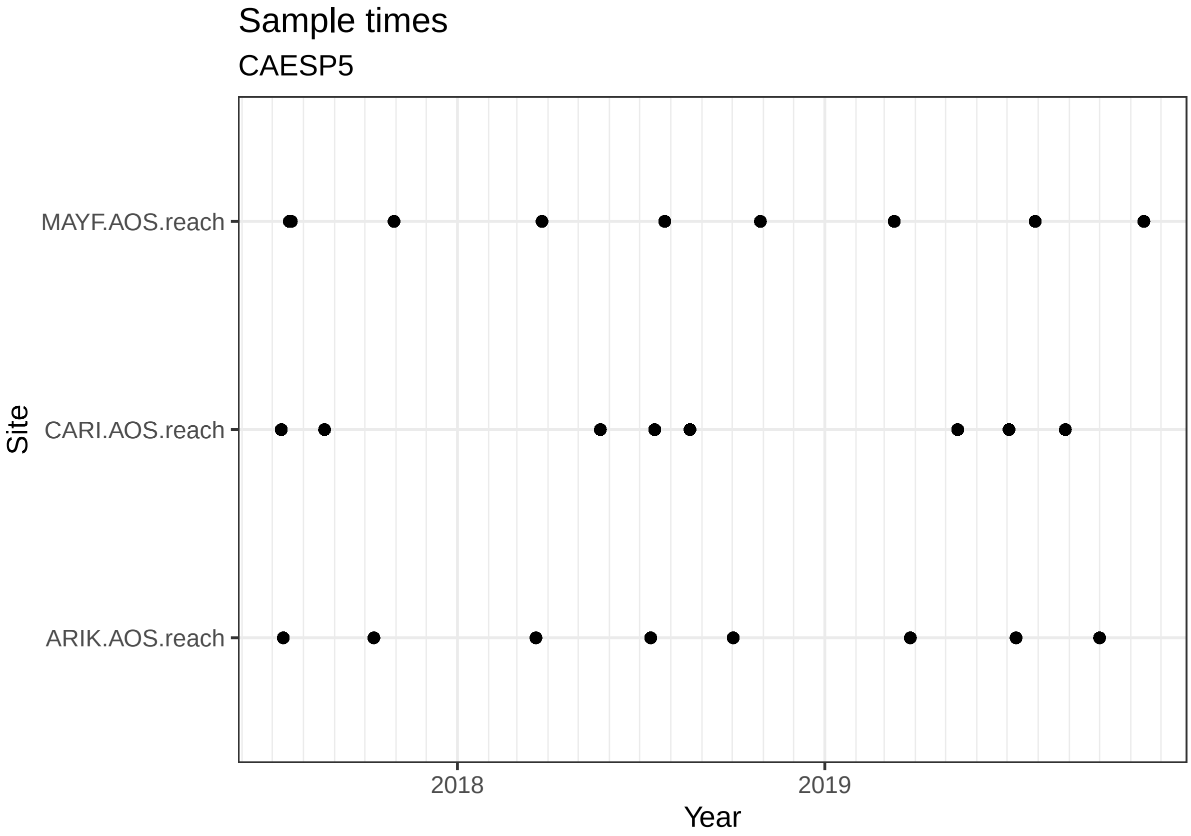

Figure 9: The months that each of the sites were sampled is recorded by a black point.

To observe how the number of taxa changes as new sites are added, the function plot_taxa_accum_sites is used. The x-axis should show the total number of sites, which is three in this example. The y-axis displays the number of unique taxa. As more sites are added, the line will start to curve and level-off, as less and less unique taxa are identified.

```
ecocomDP::plot_taxa_accum_sites(inv[[1]]$tables$observation,
                                inv[[1]]$tables$observation$location_id)
```

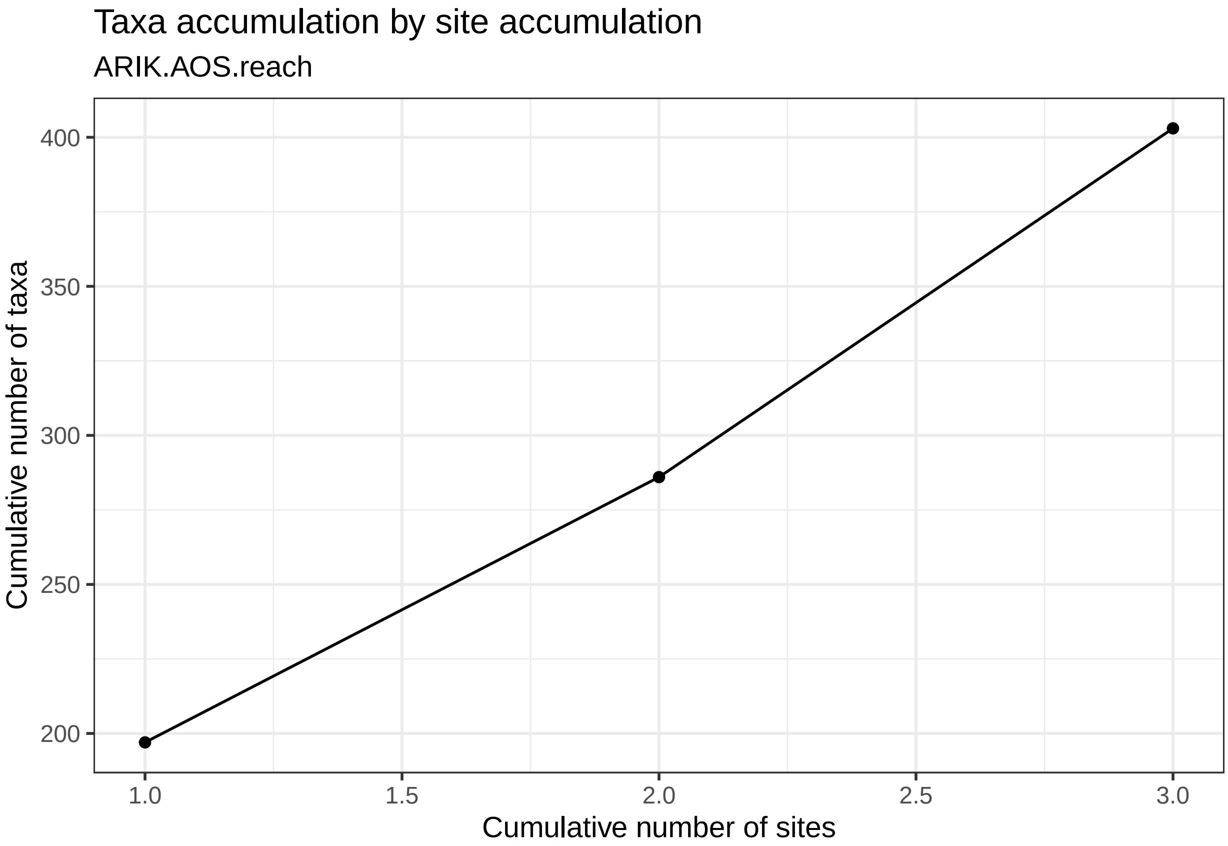

Figure 10: As the number of sites increases, the cumulative number of taxa also increases. A large number of sites would result in a curve that levels-off.

Changes in the cumulative number of taxa can also be observed over time. Using the plot_taxa_accum_time function, you can observe how many new, unique taxa are observed at each sampling time. Each site is represented by a different color, and the black line shows the summation of all sites' taxa counts.

```
ecocomDP::plot_taxa_accum_time(inv[[1]]$tables$observation,
                               inv[[1]]$tables$observation$location_id)
```

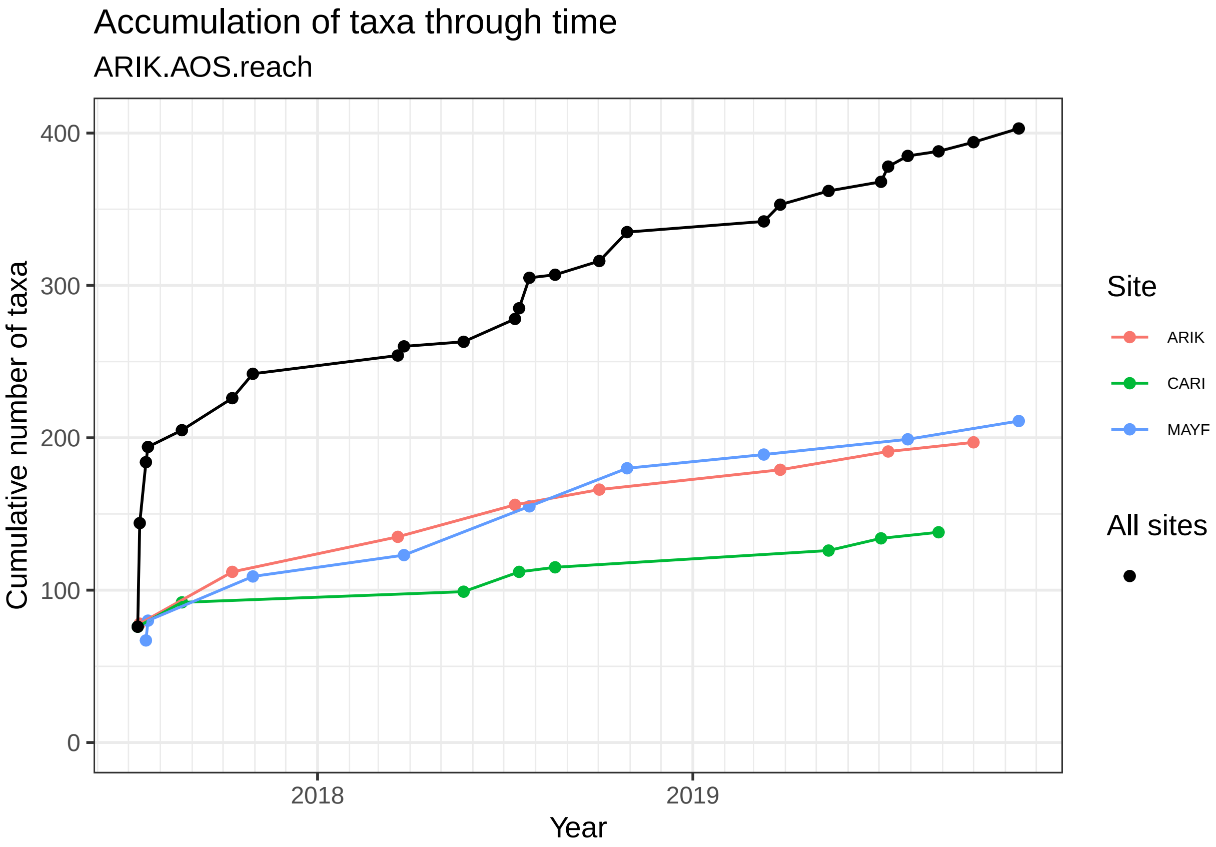

Figure 11: As time increases, the cumulative number of taxa also increases, but at a much slower rate. As more unique taxa are observed, the number of unidentified remaining taxa decreases, so there are less new ones to discover.

To see how many taxa sites have in common with each other, plot_taxa_shared_sites can be used. This function produces a matrix-like plot representing the shared taxa between all sites. Sites with higher number of shared taxa are colored red, and sites with lower number of shared taxa will be blue. The numbers along the diagonal represent the total number of taxa at each individual site.

```
ecocomDP::plot_taxa_shared_sites(inv[[1]]$tables$observation,
                                 inv[[1]]$tables$observation$location_id)
```

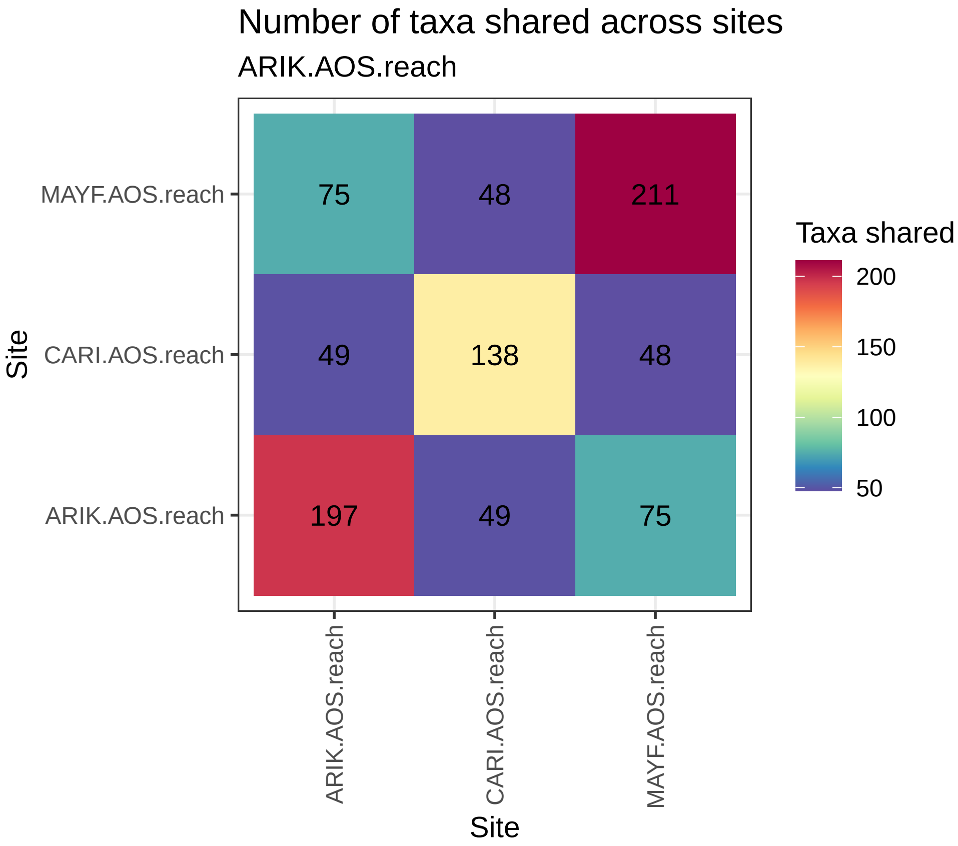

Figure 12: The sites ARIK and MAYF have the most taxa in common with each other.