BioDIGS Figure 1.

Katherine I. Cooper

2025-02-01

BioDIGS figure 1

Katherine Cooper | Enke Lab | James Madison University

The following document includes the code scripted to make Figure 1. for the BioDIGS prospective Nature Genetics paper. Each code chunk can be run in the visual mode, which is turned on in the upper left hand corner. All internal and external data can be found within the GitHub repository.

References

References for the blank map: Becker, R.A., and A.R. Wilks. 1993. "Maps in S", *AT&T Bell Laboratories Statistics Research Report*, 93.2.

Becker, R.A., and A.R. Wilks. 1995. "Constructing a Geographical Database", *AT&T Bell Laboratories Statistics Research Report*, 95.2.

US Department of Commerce, Census Bureau, County Boundary File, computer tape, available from Customer Services, Bureau of the Census, Washington DC 20233.

References for the ecoregions:

Herlihy, A.T., Paulson, S.G., Van Sickle, J., Stoddard, J.L., Hawkins, C.P., and L.L. Yuan. 2008. Striving for consistency in a national assessment: the challenges of applying a reference-condition approach at a continental scale. *Journal of the North American Benthological Society* 27(4):860-877.

Omernik, J.M. 1987. Ecoregions of the conterminous United States. Annals of the Association of American Geographers 77:118-125.

Omernik, J.M., and G.E. Griffith. 2014. Ecoregions of the conterminous United States: evolution of a hierarchical spatial framework. *Environmental Management* 54(6):1249-1266.

Part A: The map

Data processing and cleaning Load libraries

```
packages <- c("readxl",
              "ggplot2",
              "maps",
              "mapdata",
              "dplyr",
              "stringr",
```

```
              "tidyr",
              "ggrepel",
              "sf",
              "cluster",
              "ggforce",
              "patchwork",
              "ggnewscale"
)

for (pkg in packages) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
  }
  library(pkg, character.only = TRUE)
}
```

Loading the data, manipulating, and cleaning.

```
#Load BIODIGS data
BioDIGS_site_data <- read_excel(
  "C:/Users/kathe/Downloads/BioDIGS_site_data.xlsx"
  )

#Cleaning data--making the lat and long compatible with spatial mapping
BioDIGS_Cleaned <- BioDIGS_site_data %>%
  mutate(
    longitude = as.numeric(gsub("[^0-9\\.-]", "", longitude)),
    latitude  = as.numeric(gsub("[^0-9\\.-]", "", latitude)),
    longitude = as.numeric(longitude),
    latitude = as.numeric(latitude),
    long_read = gsub("\\s*\\(planned\\)\\s*", "",
                     BioDIGS_site_data$long_read), #remove planned
    long_read = trimws(long_read),
    mgmt_type_clean = ifelse(grepl("unmanaged",
                                   BioDIGS_site_data$
                                     `mgmt_type (managed vs unmanaged)`,
                                   ignore.case = TRUE),
                             "Unmanaged",
                             ifelse(grepl("managed",
                                          BioDIGS_site_data$
                                            `mgmt_type (managed vs unmanaged)`,
                                          ignore.case = TRUE),
                                    "Managed",
                                    NA)),
    #Seperate managed and unmanaged without extra terms
    across(where(is.character), ~na_if(., "NA")) #Change string NA to null values
  )
#Remove null values
BioDIGS_Cleaned <- BioDIGS_Cleaned %>%
  filter(!is.na(longitude) & !is.na(latitude))
```

Creating new variables for the site ID labels:

```r
#Set up for the looping
Current_id="Null"

iteration=0

test_list <- list()

#Long convoluted loop to make the labels.
for (i in 1:nrow(BioDIGS_Cleaned)) {
  Id_value <- substr(BioDIGS_Cleaned$site_id[i],
                     start = 1,
                     nchar(BioDIGS_Cleaned$site_id[i]) - 2)
  if (Id_value == Current_id){
    iteration <- iteration + 1
    New_row <- data.frame(
      ID_Start = Id_value,
      Label = paste0(Id_value, "1-",
                     Id_value,
                     iteration),
      Number = iteration
    )
    test_list[[i]] <- New_row
  }
  else{
    iteration = 1
    Current_id = Id_value
    New_row <- data.frame(
        ID_Start = Id_value,
        Label = paste0(Id_value,
                       iteration),
        Number = iteration
      )
      test_list[[i]] <- New_row
  }
}

final_id <- do.call(rbind,
                    test_list)

#Sorting the new data frame by the max iteration
final_id <- final_id %>%
  group_by(ID_Start) %>%
  slice_max(Number, with_ties = FALSE) %>%
  ungroup() %>%
  select(-Number)

#Joining the new label variable onto the cleaned data
BioDIGS_Cleaned <- BioDIGS_Cleaned %>%
  mutate(ID_Start = substr(site_id, 1,
                           nchar(site_id) - 2)) %>%
  left_join(final_id,
            by = "ID_Start") %>%
  select(-ID_Start)
```

```r
#Cleaning the coding environment of unnecessary variables
remove(final_id)
remove(test_list)
remove(New_row)
```

Making the labels the average coords nearby:

```r
GraphLabs <- BioDIGS_Cleaned %>%
  group_by(Label) %>%
  summarise(
    latitude  = mean(latitude,  na.rm = TRUE),
    longitude = mean(longitude, na.rm = TRUE),
    .groups = "drop"
  )

head(GraphLabs)
```

```
## # A tibble: 6 x 3
##   Label   latitude longitude
##   <chr>      <dbl>     <dbl>
## 1 A1          38.8     -77.1
## 2 B1-B24      39.3     -76.6
## 3 CL1-CL4     33.4     -81.5
## 4 CU1         36.4     -77.1
## 5 D1-D5       37.3    -108.
## 6 E1-E6       31.8    -106.
```

Filtering the data based on the DMV (DC, MD, NoVa) area

```r
#Filtering the data
BioDIGS_filtered_DMV <- BioDIGS_Cleaned %>%
  filter (
      longitude >= -78,
      longitude <= -76,
      latitude >= 38,
      latitude <= 40
  )
#Filtering the labels
DMV_Labs <- GraphLabs %>%
  filter(
    longitude >= -78,
    longitude <= -76,
    latitude >= 38,
    latitude <= 40
  )
```

Clustering for the bubbles

Managed split: Clustering for the overall using the managed v. unmanaged split:

```
set.seed(123)
coords <- BioDIGS_Cleaned[, c("longitude", "latitude")]
dist_matrix <- dist(coords)
clusters <- hclust(dist_matrix, method = "average")
BioDIGS_Cleaned$group <- cutree(clusters, h = 0.8)


bubble_data <- BioDIGS_Cleaned %>%
  group_by(group, mgmt_type_clean) %>%
  summarise(
    size = n(),
    lon = mean(longitude),
    lat = mean(latitude),
    .groups = "drop"
  )

bubble_data <- bubble_data %>%
  mutate(
    offset = ifelse(mgmt_type_clean == "Managed", 0.01, -0.01),
    lon_offset = lon + offset,
    lat_offset = lat + offset
  )
bubble_data <- na.omit(bubble_data)

bubble_data <- st_as_sf(bubble_data, coords = c("lon", "lat"), crs = 4326)

#Cleaning the environment.
remove("coords")
remove("dist_matrix")
remove("clusters")
BioDIGS_Cleaned <- subset(BioDIGS_Cleaned, select = -group)
```

Clustering for the DMV area using the managed v. unmanaged split:

```
set.seed(123)
coords <- BioDIGS_filtered_DMV[, c("longitude", "latitude")]
dist_matrix <- dist(coords)
clusters <- hclust(dist_matrix, method = "average")
BioDIGS_filtered_DMV$group <- cutree(clusters, h = 0.2)


bubble_data_DMV <- BioDIGS_filtered_DMV %>%
  group_by(group, mgmt_type_clean) %>%
  summarise(
    size = n(),
    lon = mean(longitude),
    lat = mean(latitude),
    .groups = "drop"
  )

bubble_data_DMV <- bubble_data_DMV %>%
  mutate(
```

```
    offset = ifelse(mgmt_type_clean == "Managed", 0.01, -0.01),
    lon_offset = lon + offset,
    lat_offset = lat + offset
  )
bubble_data_DMV <- na.omit(bubble_data_DMV)

#Cleaning the environment.
remove("coords")
remove("dist_matrix")
remove("clusters")
```

```
set.seed(123)
coords <- BioDIGS_Cleaned[, c("longitude", "latitude")]
dist_matrix <- dist(coords)
clusters <- hclust(dist_matrix, method = "average")
BioDIGS_Cleaned$group <- cutree(clusters, h = 0.8)


bubble_data_plain <- BioDIGS_Cleaned %>%
  group_by(group) %>%
  summarise(
    size = n(),
    lon = mean(longitude),
    lat = mean(latitude),
  ) %>%
  ungroup()

#Cleaning the environment.
remove("coords")
remove("dist_matrix")
remove("clusters")

head(bubble_data_plain)
```

Clustering with NO managed versus unmanaged split:

```
## # A tibble: 6 x 4
##   group  size    lon   lat
##   <int> <int>  <dbl> <dbl>
## 1     1    49  -76.9  39.2
## 2     2     2  -80.9  33.5
## 3     3     2  -82.1  33.3
## 4     4     1  -77.1  36.4
## 5     5     5 -108.   37.3
## 6     6     6 -106.   31.8
```

```
set.seed(123)
coords <- BioDIGS_filtered_DMV[, c("longitude", "latitude")]
```

```
dist_matrix <- dist(coords)
clusters <- hclust(dist_matrix, method = "average")
BioDIGS_filtered_DMV$group <- cutree(clusters, h = 0.2)


bubble_DMV_plain <- BioDIGS_filtered_DMV %>%
  group_by(group) %>%
  summarise(
    size = n(),
    lon = mean(longitude),
    lat = mean(latitude),
  ) %>%
  ungroup()

#Cleaning the environment.
remove("coords")
remove("dist_matrix")
remove("clusters")

head(bubble_DMV_plain)
```

Clustering for DMV with NO managed versus unmanaged split:

```
## # A tibble: 3 x 4
##   group  size   lon   lat
##   <int> <int> <dbl> <dbl>
## 1     1     1 -77.1  38.8
## 2     2    24 -76.6  39.3
## 3     3    24 -77.2  39.2
```

Graphing set-up National Lakes Assessment aggregated ecoregions. I could not get the EPA ecoregions to work again on a proper scale. Canada would distort the United States if I removed it.

```
data(stateMapEnv)
US_map <- map_data("state")

Biome_data <- st_read("C:/Users/kathe/Downloads/Aggr_Ecoregions_2015")
```

```
## Reading layer ‘Aggr_Ecoregions_2015’ from data source
##   ‘C:\Users\kathe\Downloads\Aggr_Ecoregions_2015’ using driver ‘ESRI Shapefile’
## Simple feature collection with 9 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -2356069 ymin: 272048.5 xmax: 2258225 ymax: 3172577
## Projected CRS: Albers
```

```
Converted_Biomes <- st_transform(Biome_data, crs = 4326)

#Makinhg a standardized colorscheme. Semi-coolorblind friendly.
```

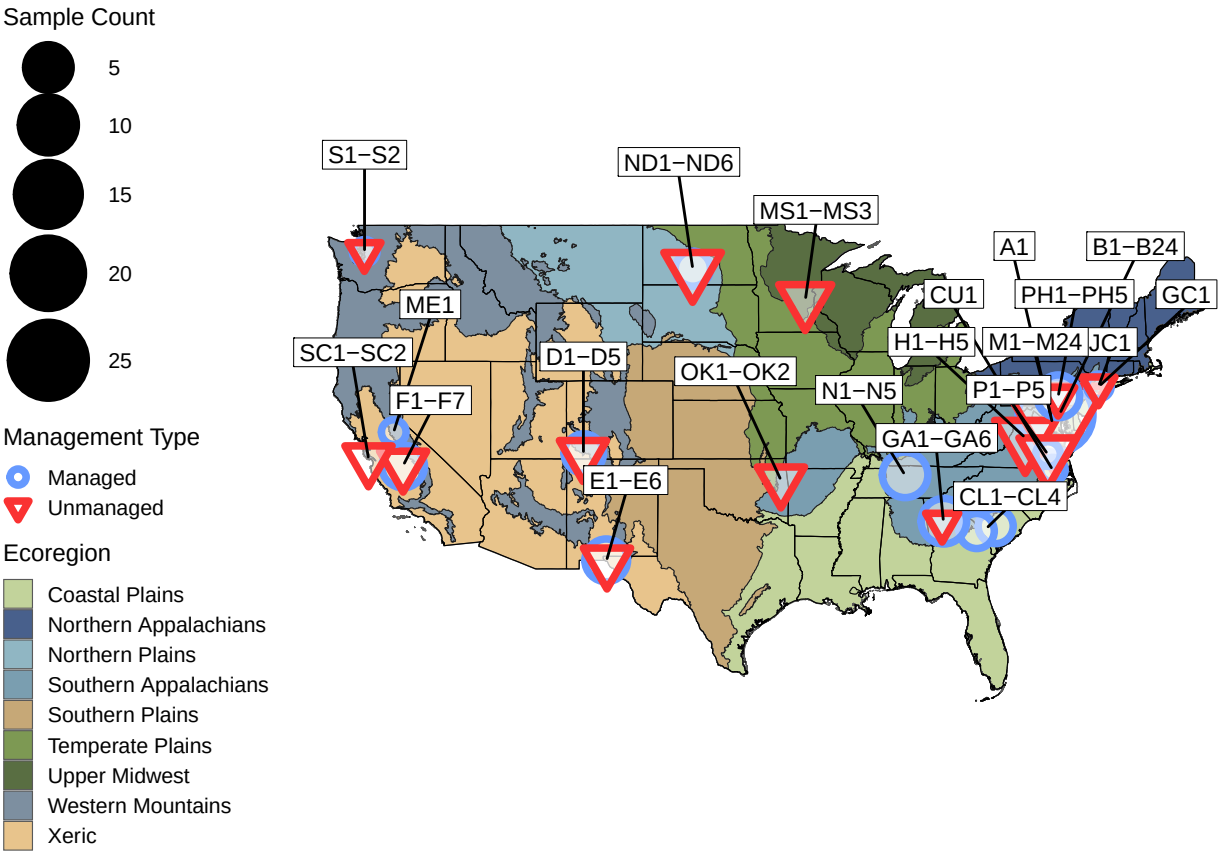
```
cols <- c(
  "Temperate Plains"      = "#7d9953",
  "Northern Appalachians" = "#48608C",
  "Northern Plains"       = "#90B6C3",
  "Southern Appalachians" = "#7A9EB0",
  "Coastal Plains"        = "#C2D39B",
  "Upper Midwest"         = "#596E42",
  "Western Mountains"     = "#7D8FA0",
  "Southern Plains"       = "#C6A877",
  "Xeric"                 = "#E8C48C"
)

#Basic plot with no data on it
Map1<-ggplot(Converted_Biomes) +
  geom_sf(aes(fill=WSA9_NAME)) +
  theme_void() + scale_fill_manual(
    values = cols,
    name = "Ecoregion") +
  geom_polygon(data = US_map,
               aes(x = long,
                   y = lat,
                   group = group),
               color = "black",
               fill = NA,
               linewidth = .2)
```

Overall map

With the managed split: Adding the bubbles to the base map

```
BubbleMap_managed <- Map1 +
  new_scale_fill()+
  geom_point(
  data = bubble_data,
  aes(x = lon_offset,
      y = lat_offset,
      size = size,
      shape = mgmt_type_clean,
      color = mgmt_type_clean),
  fill = scales::alpha("white", 0.5),
  stroke=2
) +
  scale_size_continuous(range = c(3, 12)) +
  scale_color_manual(values = c("Managed" = "#6699ff",
                                "Unmanaged" = "#fa3232"),
                     name = "Management Type") +
  scale_shape_manual(values = c("Managed" = 21,
                                "Unmanaged" = 25),
                     name = "Management Type") +
  theme_void()+
  theme(legend.position="left",
        legend.text = element_text(size = 8),
```

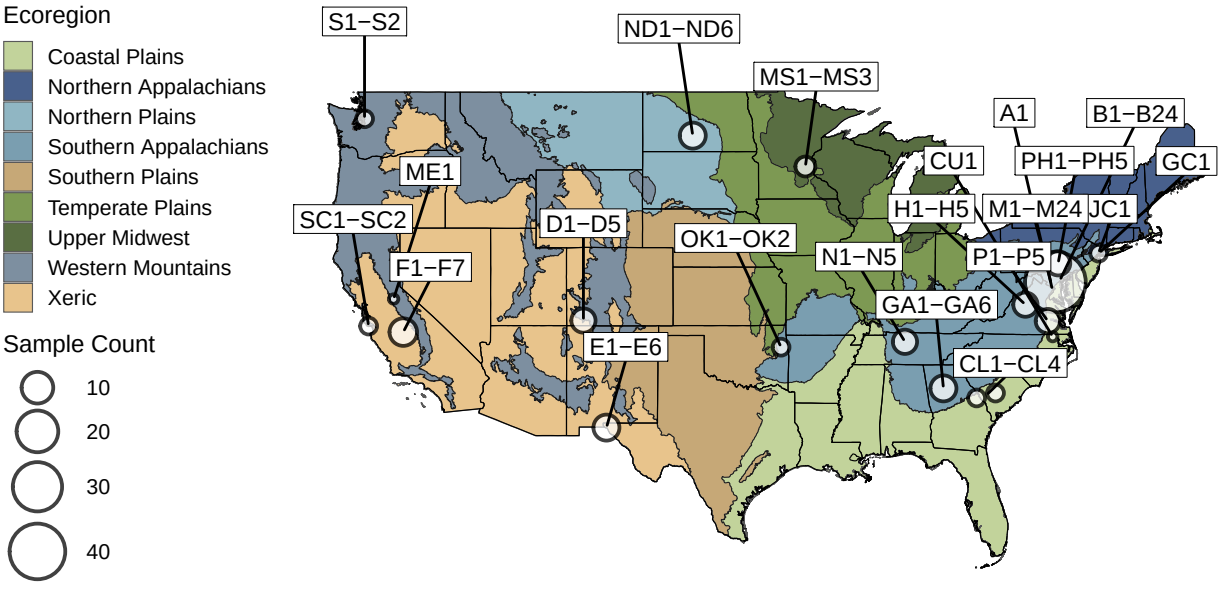
```
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))+
  labs(size="Sample Count")+
 # guides(fill = guide_legend(override.aes = list(size = 6)))+

 # scale_size_continuous(range = c(3, 12))  +

  geom_label_repel(
    data = GraphLabs,
    aes(x = longitude,
        y = latitude,
        label = Label),
    size = 3,
    nudge_y = 5,
    nudge_x = 0,
    seed = 10,
    max.overlaps = 50,
    label.padding = 0.15,
    label.size = .01,
    label.r = 0.01
  ) +
  theme(legend.position="left",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))


BubbleMap_managed
```




```
BubbleMap_plain <- Map1 +
  geom_point(
  data = bubble_data_plain,
  aes(x = lon,
      y = lat,
      size = size),
  fill ="white",
  color = "black",
  stroke=1,
  shape = 21,
  alpha = 0.75
) +
  scale_size_continuous(range = c(1, 10)) +
  theme_void()+
  theme(legend.position="left",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))+
  labs(size="Sample Count", fill = "Ecoregion")+
  guides(color = "none") +

  geom_label_repel(
    data = GraphLabs,
```

```
    aes(x = longitude,
        y = latitude,
        label = Label),
    size = 3,
    nudge_y = 5,
    nudge_x = 0,
    seed = 10,
    max.overlaps = 50,
    label.padding = 0.15,
    label.size = .01,
    label.r = 0.01
  ) +
  theme_void() +
  theme(legend.position="left",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))


BubbleMap_plain
```

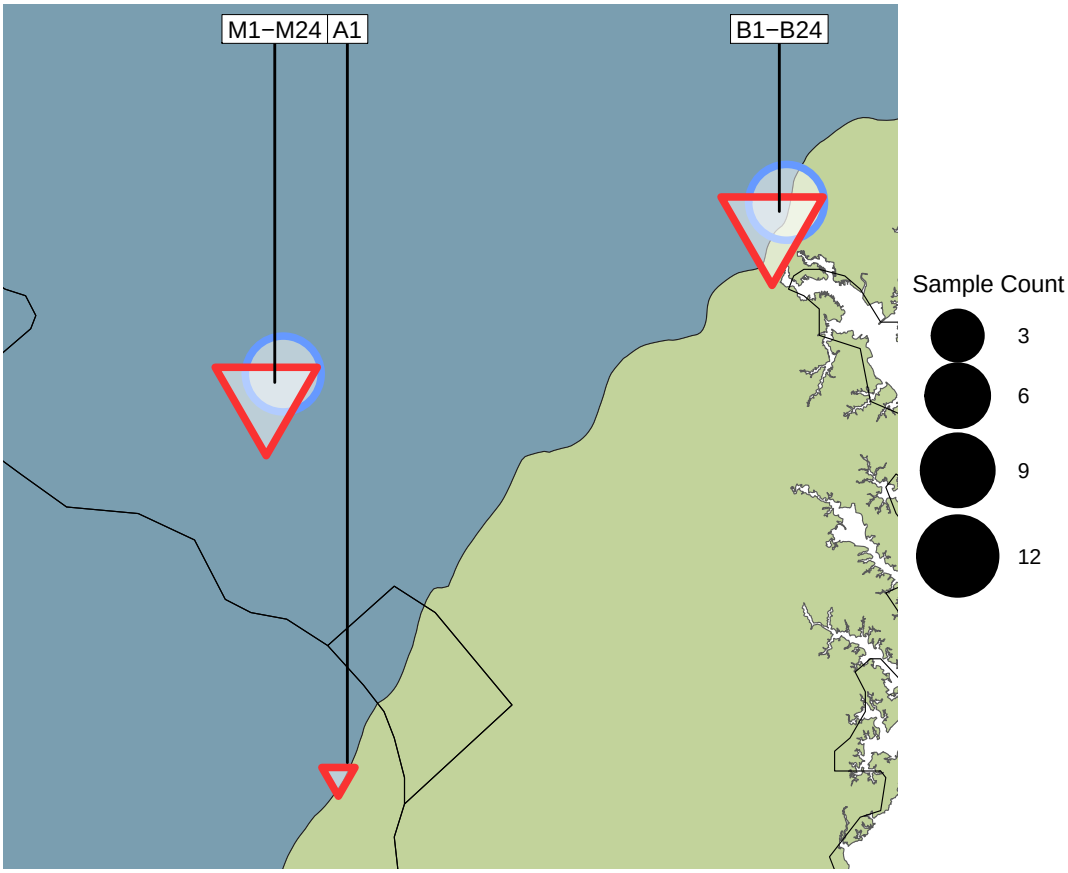
Without the managed split:



DMV cropped map

```
BubbleMap_DMVmanaged <- Map1 +
  geom_point(
  data = bubble_data_DMV,
  aes(x = lon_offset,
      y = lat_offset,
      size = size,
      shape = mgmt_type_clean,
      color = mgmt_type_clean),
  fill = scales::alpha("white", 0.5),
  stroke=2
) +
  scale_size_continuous(range = c(3, 12)) +
  scale_color_manual(values = c("Managed" = "#6699ff",
                                "Unmanaged" = "#fa3232")) +
  scale_shape_manual(values = c("Managed" = 21,
                                "Unmanaged" = 25)) +
  coord_sf(xlim = c(-77.5, -76.5),
           ylim = c(38.75, 39.50),
           expand = FALSE,
           clip = "on") + #Where the cropped is
  theme(legend.position="left",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))+
  labs(size="Sample Count")+
   guides(color = "none", fill = "none", shape = "none") +
 # guides(fill = guide_legend(override.aes = list(size = 6)))+

 # scale_size_continuous(range = c(3, 12))  +
  geom_label_repel(
    data = DMV_Labs,
    aes(x = longitude, y = latitude, label = Label),
    size = 3,
    nudge_y = 5,
    nudge_x = 0,
    seed = 10,
    max.overlaps = 50,
    label.padding = 0.15,
    label.size = .01,
    label.r = 0.01
  ) +
  theme_void() +
  theme(legend.position="right",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))


BubbleMap_DMVmanaged
```

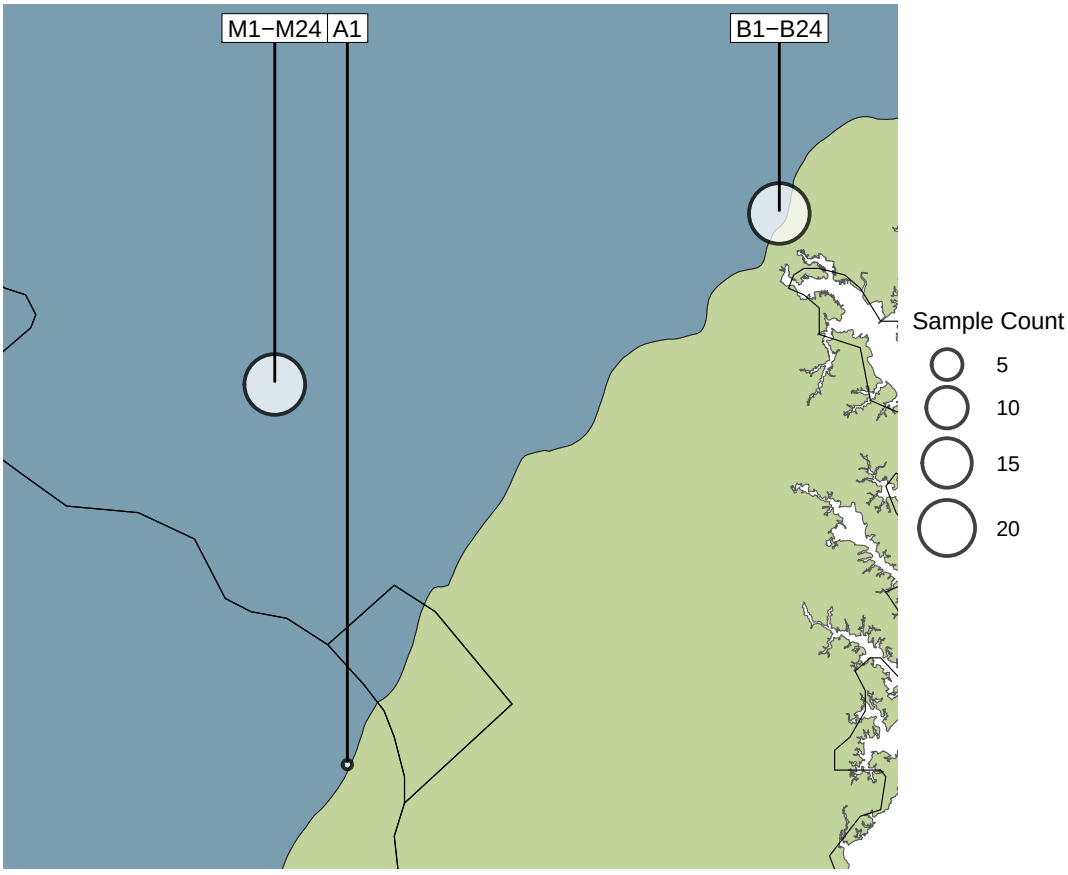
With the managed scales:




```
Ecoregion_map_inset <- Map1 +  geom_point(
  data = bubble_DMV_plain,
  aes(x = lon,
      y = lat,
      size = size),
  fill ="white",
  color = "black",
  stroke=1,
  shape = 21,
  alpha = 0.75
) +
  scale_size_continuous(range = c(1, 10)) +
  coord_sf(xlim = c(-77.5, -76.5),
           ylim = c(38.75, 39.50),
           expand = FALSE,
           clip = "on") + #Where the cropped is
  theme_void()+
  theme(legend.position="left",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))+
  labs(size="Sample Count")+
  guides(color = "none", fill = "none") +
```

```
  geom_label_repel(
    data = DMV_Labs,
    aes(x = longitude,
        y = latitude,
        label = Label),
    size = 3,
    nudge_y = 5,
    nudge_x = 0,
    seed = 10,
    max.overlaps = 50,
    label.padding = 0.15,
    label.size = .01,
    label.r = 0.01
  ) +
  theme_void() +
  theme(legend.position="right",
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 9),
        legend.key.size = unit(0.4, "cm"))

Ecoregion_map_inset
```

Without the managed/unmanaged split:



Part C: Sequencing type

Assigning yes-no Assigning Boolean variables for the sequencing type.

```
BioDIGS_Donut <- BioDIGS_Cleaned %>%
  mutate(
    Pseudo_reads = case_when(
      grepl("yes",
            short_read,
            ignore.case = TRUE) & grepl("yes",
                                        long_read,
                                        ignore.case = TRUE) ~ 1,
      grepl("yes",
            short_read,
            ignore.case = TRUE) & grepl("no",
                                        long_read,
                                        ignore.case = TRUE)  ~ 2,
      grepl("no",
            short_read,
            ignore.case = TRUE)  & grepl("yes",
                                         long_read,
                                         ignore.case = TRUE) ~ 3,
      TRUE ~ NA_real_
    ),
  )

Read_counts <- count(BioDIGS_Donut, Pseudo_reads)
Read_counts$perc <- Read_counts$n/sum(Read_counts$n)
Read_counts$ymax = cumsum(Read_counts$perc)
Read_counts$ymin = c(0, head(Read_counts$ymax, n=-1))

Read_counts$Pseudo_reads <- as.character(Read_counts$Pseudo_reads)
remove(BioDIGS_Donut)
```

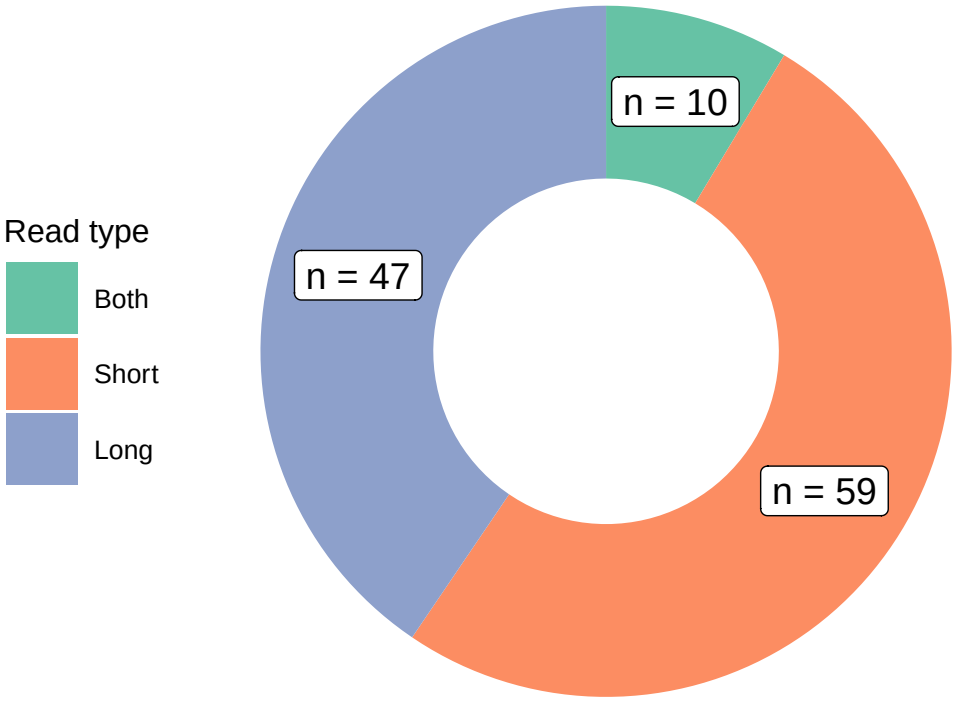
Creating the labels

```
# Compute label position
Read_counts$labelPosition <- (Read_counts$ymax + Read_counts$ymin) / 2

# Compute a good label
Read_counts$label <- paste0("n = ", Read_counts$n)
```

```
Donut_plot<-ggplot(Read_counts, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3)) +
  geom_rect(aes(fill=Pseudo_reads)) +
  coord_polar(theta="y") +
  geom_label( x=3.5,
              aes(y=labelPosition, label=label),
              size=5,
              fill = "white") +
  xlim(c(2, 4))+theme_void()+
```

```
  theme(legend.position="left",
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12),
        legend.key.size = unit(1, "cm")) +
  scale_fill_manual(
    values = c("1" = "#66c2a5", "2" = "#fc8d62", "3" = "#8da0cb"),
    labels = c("1" = "Both","2" = "Short", "3" = "Long"),
    name = "Read type"
  )

Donut_plot
```

Plotting



Part D: Soil composition

```
Soil_Makeup <- read_excel(
  "C:/Users/kathe/Downloads/BioDIGS_soil_testing_data.xlsx",
  sheet = "modified"
  )
```

```
head(Soil_Makeup)
```

Loading in the data

```
## # A tibble: 6 x 21
##   site_id As       Cd    Cr    Cu...5 Ni    Pb    Zn...8 P     K     Ca    Mg
##   <chr>   <chr>    <chr> <chr> <chr>  <chr> <chr> <chr>  <chr> <chr> <chr> <chr>
## 1 A01     4        < 0.2 10.8  5.6    4.96  38.1  22.25  7.1   57.6  207.~ 35.2~
## 2 A01     4.90000~ < 0.2 11.9  6.3    5.43  40.9~ 23.27  6.1   53.8  204.~ 33.61
## 3 B01     < 3.0    0.26  12.77 13.54  6.93  10.69 33.71  47.4  136.~ 1384~ 198.~
## 4 B02     5.04     0.53  29.02 31.48  10.75 88.21 83.35  82.21 139.~ 3062~ 276.~
## 5 B03     6.39     0.68  40.64 56.91  27.12 125.~ 160.54 32.9~ 167.~ 1792~ 234.~
## 6 B04     4.87     0.35  23.78 25.53  11.58 43.73 71.66  13.15 184.~ 1577~ 270.~
## # i 9 more variables: Mn <chr>, Zn...14 <chr>, Cu...15 <chr>, Fe <chr>,
## #   B <chr>, S <chr>, Na <chr>, Al <chr>, water_pH <chr>
```

Cleaning the data Removing "Not yet tested," "NA," "<," and any white space remaining. Then, converting the element columns to numeric from character.

```
#Removing strings from where we want numeric data
Soil_Makeup <- Soil_Makeup %>%
  mutate(across(
    where(is.character),
    ~ gsub("<", "", na_if(na_if(.x, "Not yet tested"), "NA"))
  )) %>%
  mutate(across(
    where(is.character),
    ~ trimws(.x, which = "both")
  ))

#Removing any NAs from the previous step
Soil_Makeup <- na.omit(Soil_Makeup)

#Changing the columns to numeric from character

Soil_Makeup <- Soil_Makeup %>%
  mutate(across(
    2:ncol(Soil_Makeup),
    ~ as.numeric(.x)
  ))
```

Pivoting from wide format to long format. This is the format needed for histograms. Wide format is good for heatmaps.

```
Soil_Long <- Soil_Makeup %>%
  pivot_longer(
    cols = 2:ncol(Soil_Makeup),
    names_to = "Element",
    values_to = "Values"
  )

head(Soil_Long)
```

```
## # A tibble: 6 x 3
##   site_id Element Values
##   <chr>   <chr>    <dbl>
## 1 A01     As        4
## 2 A01     Cd        0.2
## 3 A01     Cr       10.8
## 4 A01     Cu...5    5.6
## 5 A01     Ni        4.96
## 6 A01     Pb       38.1
```

Soil long does not include the managed versus unmanaged data. This can be mapped to the Soil_Long object with the site ID.

```
Temp_obj <- BioDIGS_Cleaned[c("site_id", "mgmt_type_clean")]

Soil_Long <- left_join(Soil_Long, Temp_obj, by = "site_id")

# Cleaning the environment
remove(Temp_obj)

#View the top of the new frame
head(Soil_Long)
```

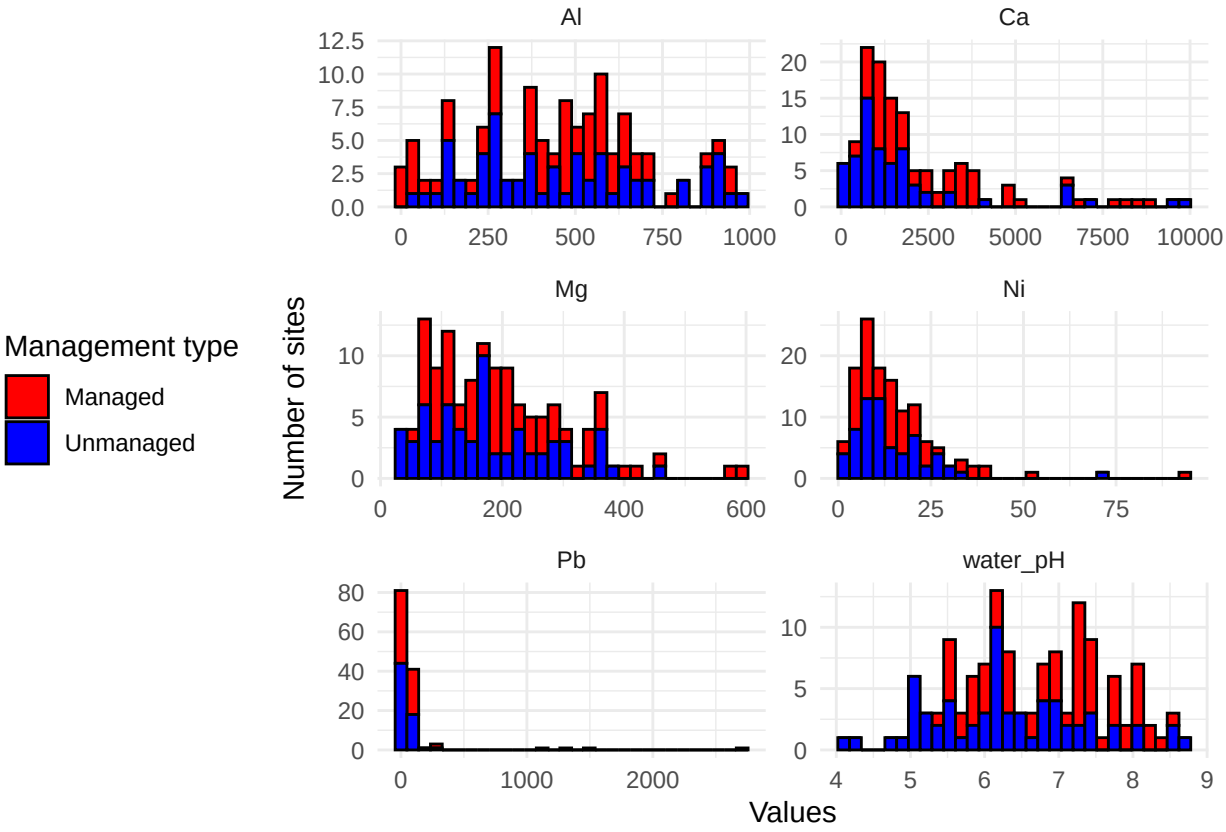
```
## # A tibble: 6 x 4
##   site_id Element Values mgmt_type_clean
##   <chr>   <chr>    <dbl> <chr>
## 1 A01     As        4    Unmanaged
## 2 A01     Cd        0.2  Unmanaged
## 3 A01     Cr       10.8  Unmanaged
## 4 A01     Cu...5    5.6  Unmanaged
## 5 A01     Ni        4.96 Unmanaged
## 6 A01     Pb       38.1  Unmanaged
```

Filtering by element Filter to keep the requested elements

```
elements <- c("Al", "Ar", "Ca", "water_pH", "Mg", "Pb", "Ni")
Soil_long_filtered <- Soil_Long[Soil_Long$Element %in% elements, ]

# Cleaning the environment
remove(elements)
```

Making the histogram Faceting around the element

```
ggplot(Soil_long_filtered, aes(x=Values, fill = mgmt_type_clean)) +
  geom_histogram(colour = "black",
                 stroke = 0.1) +
  facet_wrap("Element", scales = "free", ncol=2) +
  scale_fill_manual(values = c(Managed = "red", Unmanaged = "blue")) +
  labs(y= "Number of sites", fill = "Management type") +
  theme_minimal() +
  theme(legend.position = "left")
```


Management type
Managed
Unmanaged
Al
Ca
Mg
Ni
Pb
water_pH
Number of sites
Values