SEACAR Nekton Analysis

Last compiled on 29 June, 2022

# Important Notes

All scripts and outputs can be found on the SEACAR GitHub repository:

<https://github.com/FloridaSEACAR/SEACAR_Panzik>

# Libraries and Settings

Loads libraries used in the script. Loads the Segoe UI font for use in the figures. The inclusion of scipen option limits how frequently R defaults to scientific notation. Sets default settings for displaying warning and messages in created document, and sets figure dpi.

library(knitr)  
library(data.table)  
library(dplyr)  
library(lubridate)  
library(ggplot2)  
library(scales)  
library(tidyr)  
library(gridExtra)  
library(tidyverse)  
windowsFonts(`Segoe UI` = windowsFont('Segoe UI'))  
options(scipen=999)  
opts\_chunk$set(warning=FALSE, message=FALSE, dpi=200)

# File Import

Imports file that is determined in the SEACAR\_Nekton\_Render.R script.

The command fread is used because of its improved speed while handling large data files. Only columns that are used by the script are imported from the file, and are designated in the select input.

The script then gets the name of the parameter as it appears in the data file and units of the parameter.

data <- fread(file\_in, sep="|", header=TRUE, stringsAsFactors=FALSE,  
 # select=c("ManagedAreaName", "ProgramID", "ProgramName",  
 # "ProgramLocationID", "SampleDate", "Year", "Month",  
 # "RelativeDepth", "ActivityType", "ParameterName",  
 # "ResultValue", "ParameterUnits", "ValueQualifier",  
 # "SEACAR\_QAQCFlagCode", "Include"),  
 na.strings="")  
parameter <- unique(data$ParameterName)  
if(parameter=="Presence"){  
 parameter <- "Species Richness"  
 param\_name <- "SpeciesRichness"  
}  
unit <- unique(data$ParameterUnits)

# Data Filtering

Documentation on database filtering is provided here: [SEACAR Documentation- Analysis Filters and Calculations.docx](https://github.com/FloridaSEACAR/SEACAR_Panzik/blob/main/SEACAR%20Documentation%20-%20Analysis%20Filters%20and%20Calculations.docx)

The filtering that is performed by the script at this point removes rows that are missing values for ResultValue and EffortCorrection\_100m2, and removes any EffortCorrection\_100m2 that is 0 because it will cause an infinite number when determining Species Richness.

A group of unique ManagedAreaName, ProgramID, ProgramName, ProgramLocationID, SampleDate, and GearSize\_m are being considered a “reference” for measurement. For each “reference”, the number of observed species is summed and then divided by the EffortCorrection\_100m2to determine the Species Richness per 100 square meters.

The ManagedAreaName values from the data are actually shortened versions, and are merged with the full versions. The species richness data is then written to a file. And the list of Managed Areas with observations is stored.

data <- data[!is.na(data$EffortCorrection\_100m2),]  
data <- data[data$EffortCorrection\_100m2!=0,]  
data <- data[!is.na(data$ResultValue),]  
  
data <- data %>%  
 group\_by(ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,  
 SampleDate, GearSize\_m) %>%  
 summarise(ParameterName=parameter,  
 Year=unique(Year), Month=unique(Month),  
 N\_Species=sum(ResultValue),  
 EffortCorrection\_100m2=unique(EffortCorrection\_100m2),  
 SpeciesRichness=N\_Species/unique(EffortCorrection\_100m2))  
  
setnames(data, c("ManagedAreaName"), c("ShortName"))  
  
data <- merge.data.frame(MA\_All[,c("AreaID", "ManagedAreaName", "ShortName")],  
 data, by="ShortName", all=TRUE)  
  
data$ShortName <- NULL  
  
fwrite(data, paste0(out\_dir,"/Nekton\_", param\_name, "\_UsedData.txt"), sep="|")  
  
data$SampleDate <- as.Date(data$SampleDate)  
  
MA\_Include <- unique(data$ManagedAreaName[!is.na(data$N\_Species)])  
MA\_Include <- MA\_Include[order(MA\_Include)]  
n <- length(MA\_Include)

# Managed Area Statistics

Gets summary statistics for each managed area. Uses piping from dplyr package to feed into subsequent steps. The following steps are performed:

1. Group data that have the same ManagedAreaName, Year, Month, and GearSize\_m.
   * Second summary statistics do not use the Month grouping and are only for ManagedAreaName and Year.
   * Third summary statistics do not use Year grouping and are only for ManagedAreaName and Month
2. For each group, provide the following information: Parameter Name (ParameterName), Number of Entries (N\_Data), Lowest Value (Min), Largest Value (Max), Median, Mean, Standard Deviation, and a list of all Program IDs included in these measurements.
3. Sort the data in ascending (A to Z and 0 to 9) order based on ManagedAreaName then Year then Month
4. Write summary stats to a pipe-delimited .txt file in the output directory
   * [Click this text to open Git directory with output files](https://github.com/FloridaSEACAR/SEACAR_Panzik/tree/main/output/WQ)

MA\_YM\_Stats <- data %>%  
 group\_by(AreaID, ManagedAreaName, Year, Month, GearSize\_m) %>%  
 summarize(ParameterName=parameter,  
 N\_Data=length(na.omit(SpeciesRichness)),  
 Min=min(SpeciesRichness),  
 Max=max(SpeciesRichness),  
 Median=median(SpeciesRichness),  
 Mean=mean(SpeciesRichness),  
 StandardDeviation=sd(SpeciesRichness),  
 ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),  
 collapse=', '))  
MA\_YM\_Stats <- as.data.table(MA\_YM\_Stats[order(MA\_YM\_Stats$ManagedAreaName,  
 MA\_YM\_Stats$Year,  
 MA\_YM\_Stats$Month,  
 MA\_YM\_Stats$GearSize\_m), ])  
fwrite(MA\_YM\_Stats, paste0(out\_dir,"/Nekton\_", param\_name,  
 "\_ManagedArea\_YearMonth\_Stats.txt"), sep="|")  
rm(MA\_YM\_Stats)  
  
MA\_Y\_Stats <- data %>%  
 group\_by(AreaID, ManagedAreaName, Year, GearSize\_m) %>%  
 summarize(ParameterName=parameter,  
 N\_Data=length(na.omit(SpeciesRichness)),  
 Min=min(SpeciesRichness),  
 Max=max(SpeciesRichness),  
 Median=median(SpeciesRichness),  
 Mean=mean(SpeciesRichness),  
 StandardDeviation=sd(SpeciesRichness),  
 ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),  
 collapse=', '))  
MA\_Y\_Stats <- as.data.table(MA\_Y\_Stats[order(MA\_Y\_Stats$ManagedAreaName,  
 MA\_Y\_Stats$Year,  
 MA\_Y\_Stats$GearSize\_m), ])  
fwrite(MA\_Y\_Stats, paste0(out\_dir,"/Nekton\_", param\_name,  
 "\_ManagedArea\_Year\_Stats.txt"), sep="|")  
  
MA\_M\_Stats <- data %>%  
 group\_by(AreaID, ManagedAreaName, Month, GearSize\_m) %>%  
 summarize(ParameterName=parameter,  
 N\_Data=length(na.omit(SpeciesRichness)),  
 Min=min(SpeciesRichness),  
 Max=max(SpeciesRichness),  
 Median=median(SpeciesRichness),  
 Mean=mean(SpeciesRichness),  
 StandardDeviation=sd(SpeciesRichness),  
 ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),  
 collapse=', '))  
MA\_M\_Stats <- as.data.table(MA\_M\_Stats[order(MA\_M\_Stats$ManagedAreaName,  
 MA\_M\_Stats$Month,  
 MA\_M\_Stats$GearSize\_m), ])  
fwrite(MA\_M\_Stats, paste0(out\_dir,"/Nekton\_", param\_name,  
 "\_ManagedArea\_Month\_Stats.txt"), sep="|")  
rm(MA\_M\_Stats)  
  
MA\_Ov\_Stats <- data %>%  
 group\_by(AreaID, ManagedAreaName, GearSize\_m) %>%  
 summarize(ParameterName=parameter,  
 N\_Years=length(unique(na.omit(Year))),  
 EarliestYear=min(Year),  
 LatestYear=max(Year),  
 N\_Data=length(na.omit(SpeciesRichness)),  
 Min=min(SpeciesRichness),  
 Max=max(SpeciesRichness),  
 Median=median(SpeciesRichness),  
 Mean=mean(SpeciesRichness),  
 StandardDeviation=sd(SpeciesRichness),  
 ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),  
 collapse=', '))  
MA\_Ov\_Stats <- as.data.table(MA\_Ov\_Stats[order(MA\_Ov\_Stats$ManagedAreaName,  
 MA\_Ov\_Stats$GearSize\_m), ])  
fwrite(MA\_Ov\_Stats, paste0(out\_dir,"/Nekton\_", param\_name,  
 "\_ManagedArea\_Overall\_Stats.txt"), sep="|")  
MA\_Ov\_Stats <- MA\_Ov\_Stats[!is.na(MA\_Ov\_Stats$EarliestYear), ]

# Appendix I: Managed Area Species Richness

The plots shown here are the species richness for each managed area with a yearly average, separated by gear size.

1. Set common plot theme.
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Set the plot type as a point plot with the size of the points
4. Add the linear trend
5. Create the title, x-axis, y-axis, and color fill labels
6. Set the y and x limits
7. Apply common plot theme
8. Create file name to save figure
9. Save figure as png file

plot\_theme <- theme\_bw() +  
 theme(text=element\_text(family="Segoe UI"),  
 title=element\_text(face="bold"),  
 plot.title=element\_text(hjust=0.5, size=14, color="#314963"),  
 plot.subtitle=element\_text(hjust=0.5, size=10, color="#314963"),  
 axis.title.x = element\_text(margin = margin(t = 5, r = 0,  
 b = 10, l = 0)),  
 axis.title.y = element\_text(margin = margin(t = 0, r = 10,  
 b = 0, l = 0)),  
 axis.text=element\_text(size=10),  
 axis.text.x=element\_text(face="bold", angle = 60, hjust = 1),  
 axis.text.y=element\_text(face="bold"))  
  
colorlist <- c("#00374f",  
 "#004d68",  
 "#006481",  
 "#007c99",  
 "#0094b0",  
 "#00aec6",  
 "#00c9db",  
 "#00e4ee",  
 "#00ffff")  
  
if(n==0){  
 print("There are no monitoring locations that qualify.")  
} else {  
 for (i in 1:n) {  
 plot\_data <- MA\_Y\_Stats[MA\_Y\_Stats$ManagedAreaName==MA\_Include[i]]  
 t\_max <- max(MA\_Ov\_Stats$LatestYear[MA\_Ov\_Stats$ManagedAreaName==  
 MA\_Include[i]])  
 t\_min <- min(MA\_Ov\_Stats$EarliestYear[MA\_Ov\_Stats$ManagedAreaName==  
 MA\_Include[i]])  
 t <- t\_max-t\_min  
   
 if(t>=30){  
 brk <- -10  
 }else if(t<30 & t>=10){  
 brk <-5  
 }else if(t<10 & t>=4){  
 brk <- -2  
 }else if(t<4){  
 brk <- -1  
 }  
   
 p1 <- ggplot(data=plot\_data, group=as.factor(GearSize\_m)) +  
 geom\_point(aes(x=Year, y=Mean, fill=as.factor(GearSize\_m)), shape=21,  
 size=3, color="#333333", alpha=0.75) +  
 geom\_line(aes(x=Year, y=Mean, color=as.factor(GearSize\_m)),  
 size=1.2, alpha=0.75) +  
 labs(title=MA\_Include[i],  
 subtitle="Species Richness",  
 x="Year", y="Number of Species per 100 square meters",  
 fill="Gear Size (m)", color="Gear Size (m)") +  
 scale\_x\_continuous(limits=c(t\_min-0.25,  
 t\_max+0.25),  
 breaks=seq(t\_max,  
 t\_min,  
 brk)) +  
 plot\_theme  
   
 outname <- paste0("Nekton\_", gsub(" ", "", MA\_Include[i]), "\_",  
 param\_name, ".png")  
 png(paste0(out\_dir, "/Figures/", outname),  
 width = 8,  
 height = 4,  
 units = "in",  
 res = 200)  
 rm(plot\_data)  
   
 print(p1)  
 dev.off()  
   
   
 ResultTable <- MA\_Ov\_Stats[MA\_Ov\_Stats$ManagedAreaName==MA\_Include[i],]  
 ResultTable <- ResultTable[,-c("AreaID", "ManagedAreaName",  
 "ProgramIDs")]  
 t1 <- ggtexttable(ResultTable, rows = NULL,  
 theme=ttheme(base\_size=7))  
 print(ggarrange(p1, t1, ncol=1, heights=c(0.85, 0.15)))  
   
 cat("\n \n \n \n")  
 }  
}











