V1075 Monte Carlo

Matthew Allen

December 2023

## Introduction

### Here I demonstrate how one might run a Monte Carlo on 18O-derived temperature proxies in order to estimate uncertainty.

In this example we apply a dual-taxon temperature proxy sensu Fricke and Wing (2007). We’ll use 18O data that I generated from the Albian (mid-Cretaceous) Oklahoma Museum of Natural History V1075 vertebrate microfossil assemblage from the Cloverly Formation of Montana.

You can replace this input data with your data of interest. You’ll need 18O data from an aquatic turtle and a freshwater fish, both from the same assemblage. You will also need the 18O of the NIST120c standards that were run with your samples. The turtle and fish .csv file will need to be formatted with mean 18O for each specimen, as well as the eco\_type/taxon of each specimen.

## Step 1: Load packages and data

## !!! I want to source the data directly from GitHub, but Rmarkdown is giving me trouble !!!

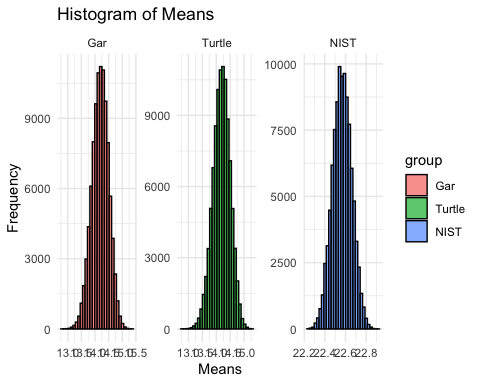
# Packages  
packages <- c("dplyr", "purrr", "ggplot2", "Rcurl")  
  
# install.packages(packages)  
  
library(dplyr)  
library(purrr)  
library(ggplot2)  
# library(Rcurl)  
  
# Data  
 # Call d18Op data from GitHub, cleaned and averaged per specimen (V1075\_BySpec.csv)  
 # samples\_URL <- getURL("https://raw.githubusercontent.com/mattgeo1990/1075\_Vertebrate\_d18Op/main/Data/V1075\_GarTurtle")  
 # V1075\_GarTurtle <- read.csv(text = samples\_URL)  
  
 # Call NIST120c data from Github, compiled from 2 TC-EA runs  
 # standards\_URL <-"https://raw.githubusercontent.com/mattgeo1990/1075\_Vertebrate\_d18Op/main/Data/V1075\_NIST120c.csv"  
 # NIST120c <- read.csv(standards\_URL)  
  
#   
setwd("/Users/allen/Documents/GitHub/1075\_Vertebrate\_d18Op/Data")  
V1075\_GarTurtle <- read.csv("V1075\_GarTurtle.csv")  
NIST120c <- read.csv("V1075\_NIST120c.csv")

## Step 2: Resampling the isotope data

Here we are resampling with replacement and taking the mean of each resample.

Let’s plot a histogram to look at our resample means.

# Set number of Monte Carlo repetitions   
 nMCrepetitions <- 1e5  
  
# Subset V1075\_GarTurtle into gar and turtle matrices  
 gar <- subset(V1075\_GarTurtle, eco\_type = "Fish")  
 turtle <- subset(V1075\_GarTurtle, eco\_type = "Aquatic Turtle")  
  
# resample  
  
 synth\_gar <- data\_frame(num = 1:nMCrepetitions) %>%   
 group\_by(num) %>%   
 mutate(means = mean(sample(gar$d18O, replace = TRUE)))   
  
 synth\_turtle <- data\_frame(num = 1:nMCrepetitions) %>%   
 group\_by(num) %>%   
 mutate(means = mean(sample(turtle$d18O, replace = TRUE)))   
  
 synth\_NIST <- data\_frame(num = 1:nMCrepetitions) %>%   
 group\_by(num) %>%   
 mutate(means = mean(sample(NIST120c$d.18O.16O, replace = TRUE)))   
  
 # Plot histograms of each distribution  
 # Combine data frames into one  
 combined\_data <- rbind(  
 data.frame(group = "Gar", values = synth\_gar$means),  
 data.frame(group = "Turtle", values = synth\_turtle$means),  
 data.frame(group = "NIST", values = synth\_NIST$means)  
 )  
  
 # Set order of facets  
 combined\_data$group <- factor(combined\_data$group, levels = c("Gar", "Turtle", "NIST"))  
  
 # Plotting with ggplot2 using facet\_wrap  
 ggplot(combined\_data, aes(x = values, fill = group)) +  
 geom\_histogram(position = "identity", alpha = 0.7, bins = 30, color = "black") +  
 labs(title = "Histogram of Means",  
 x = "Means",  
 y = "Frequency") +  
 theme\_minimal() +  
 facet\_wrap(~group, scales = "free")

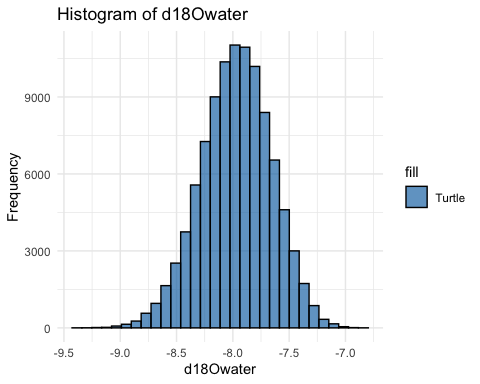


## Step 3: Calculate water oxygen isotope values from simulated turtle samples

Note that we are only using data from an aquatic turtle genus Glyptops spp. It is important to choose a turtle taxon that is thought to have been mostly or entirely aquatic in habit.

We’ll do this using the relationship between environmental water and turtle bone defined by Barrick et al. (1999).

# calculate water  
  
 # create function  
 turtlewater <- function(synthmeans\_turtle){  
 1.01 \*(synthmeans\_turtle) - 22.3 #Barrick et al. (1999)  
 }  
  
 # run turtlewater on synth\_turtle  
 synth\_turtle <- synth\_turtle %>%  
 mutate(d18Owater = turtlewater(means))  
  
# Plot histogram of d18Owater  
 ggplot(synth\_turtle, aes(x = d18Owater, fill = "Turtle")) +  
 geom\_histogram(position = "identity", alpha = 0.7, bins = 30, color = "black") +  
 labs(title = "Histogram of d18Owater",  
 x = "d18Owater",  
 y = "Frequency") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = c("#1f78b4"))



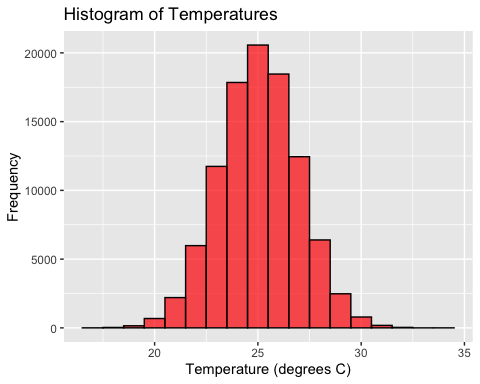
# CI, mean of turtle d18Owater  
 CIsetup\_water <- sort(synth\_turtle$d18Owater)  
   
 # Calculate the lower and upper percentiles for the middle 95%  
 lower\_percentile\_water <- quantile(CIsetup\_water, 0.025)  
 upper\_percentile\_water <- quantile(CIsetup\_water, 0.975)  
   
 # Subset the middle 95% of the data  
 subset\_CIsetup\_water <- CIsetup\_water[CIsetup\_water >= lower\_percentile\_water & CIsetup\_water <= upper\_percentile\_water]  
   
 # Take mean  
 mean\_turtlewater <- round(mean(subset\_CIsetup\_water), 1)  
 lowCI\_water <- round(abs(mean\_turtlewater - lower\_percentile\_water), 1)  
 highCI\_water <- round(abs(mean\_turtlewater - upper\_percentile\_water), 1)  
 cat(mean\_turtlewater, "+", highCI\_water, "/", "-", lowCI\_water)

## -8 + 0.6 / - 0.6

## Step 4: Calculate temperatures

We’ll use the relationship between 18Ophosphate and temperature defined originally by Longinelli & Nuti (1973) and refined by Puc'eat et al. (2010)

# calculate temps  
 # define temp function  
 TempFun <- function(d18Ofish, NISTmean, d18Owater) {  
 temp <- 118.7 - 4.22\*((d18Ofish +(22.6 - NISTmean)) - d18Owater)   
 }  
  
 # run TempFun over all the synthetic means  
 synth\_temps <- TempFun(d18Ofish = synth\_gar$means,   
 NISTmean = synth\_NIST$means,  
 d18Owater = synth\_turtle$d18Owater)  
   
# Plot the temperatures  
 ggplot(data = data.frame(temperature = synth\_temps), aes(x = temperature)) +  
 geom\_histogram(binwidth = 1, fill = c("#ff0000"), color = "black", alpha = 0.7) +  
 labs(title = "Histogram of Temperatures",  
 x = "Temperature (degrees C)",  
 y = "Frequency")



## Step 5: Confidence Intervals

Finally, we’ll construct confidence intervals around the mean of our temperature estimates.

# Construct confidence intervals  
 CIsetup <- sort(synth\_temps)  
   
# Calculate the lower and upper percentiles for the middle 95%  
 lower\_percentile <- quantile(CIsetup, 0.025)  
 upper\_percentile <- quantile(CIsetup, 0.975)  
   
# Subset the middle 95% of the data  
 subset\_CIsetup <- CIsetup[CIsetup >= lower\_percentile & CIsetup <= upper\_percentile]  
   
# Take mean  
 mean\_temp <- round(mean(subset\_CIsetup), 1)  
 lowCI <- round(abs(mean\_temp - lower\_percentile), 1)  
 highCI <- round(abs(mean\_temp - upper\_percentile), 1)  
 cat(mean\_temp, "+", highCI, "/", "-", lowCI)

## 25 + 3.8 / - 3.7