Data Visualization and Normalization of Results of Carbon From Stable Isotope Analysis

German Torres

May 2019

1 Abstract

Lipids in muscle tissue have an negative effect on the readings of Carbon during an stable isotope analysis, to counter this a pre-treatment is needed that normalizes the values, however doing this affects the accuracy of the readings for different elements. This program addresses the issue by utilizing a mathematical approach that corrects the Carbon values without having to do a pre-treatment. Also the program is User-Friendly to ensure that any user that utilizes it can be able to correct their data without having to deal with the code. The program returns the corrected values in a graphical form and it also visualizes the original sampling sites where the samples were taken.

2 Introduction

Isotopes are alternative versions of elements. They have the same properties, however their molecular weight and natural availability differ. Stable isotopes refer to the naturally occurring versions of these elements that do not decay over time. The flux of macro nutrients like Nitrogen and Carbon play a key role in the sustaining of any ecosystem(Pinnegar et al). Stable isotope analysis is a tool that measures the ratios between normal elemental molecules and their stable isotopic version. The analysis can give information about habitat complexity, food-web interactions and nutrient availability.

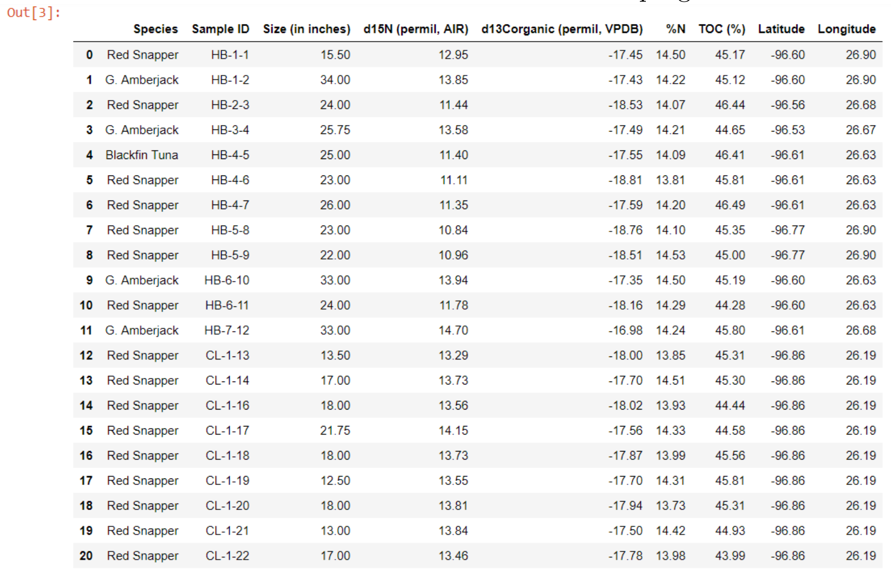
Although stable isotope analysis provides a novel tool for ecology it also has its drawbacks. One of the main issues in the analysis of organic tissues is the need of pre-treatments like acid treatment and lipid extraction to improve the quality of the readings of a specific element (Mintenback et al 2008). However this proves to be quite expensive in large numbers and also, while it increases the accuracy of the readings for a specific isotope it usually has a negative impact on the readings of the other elements. Mathematical correction for have been proposed to counterbalance these misreadings. Until now there was not a program that addressed the issue, although it has its limitations and can be improved it provides the services necessaries to be useful.

3 Methods

As the utilization of Python programming in environmental sciences keeps increasing also new ways to approach a problem are created. Before doing any programming it was important to establish what mathematical model would be utilized to address the problem. The mathematical formula for normalization chosen is proposed in Post et al (2007) where it is established to be used specifically for marine organisms. The formula is the following: $\delta^{13}C_{normalized} = \delta^{13}C_{untreated} - 3.32 + 0.99 * C : N$, it should be noted that this formula would only work if the initial analysis evaluated the isotope ratios of both Carbon and Nitrogen, which is usually done in these studies.

First off, the program prompts for the name of the file where the data to correct is to be analyzed, this is made possible utilizing the Pandas' library to be able to read the excel file while storing the data in a single variable.Next, the program also asks the user for the names of the rest of the columns (Carbon column, Nitrogen column and Coordinates columns) and it stores the data of the columns as an array of numbers. Then the program returns a table with the original data. Pandas was chosen to do the mathematical correction rather

Figure 1: Initial table containing the original data from the selected Excel spreadsheet. The d13Corganic (permil, VPDB) column is the one that contains the untreated Carbon ratios. This is the data that the program will correct.



Out[3]:

	Species	Sample ID	Size (in inches)	d15N (permil, AIR)	d13Corganic (permil, VPDB)	%N	TOC (%)	Latitude	Longitude
0	Red Snapper	HB-1-1	15.50	12.95	-17.45	14.50	45.17	-96.60	26.90
1	G. Amberjack	HB-1-2	34.00	13.85	-17.43	14.22	45.12	-96.60	26.90
2	Red Snapper	HB-2-3	24.00	11.44	-18.53	14.07	46.44	-96.56	26.68
3	G. Amberjack	HB-3-4	25.75	13.58	-17.49	14.21	44.65	-96.53	26.67
4	Blackfin Tuna	HB-4-5	25.00	11.40	-17.55	14.09	46.41	-96.61	26.63
5	Red Snapper	HB-4-6	23.00	11.11	-18.81	13.81	45.81	-96.61	26.63
6	Red Snapper	HB-4-7	26.00	11.35	-17.59	14.20	46.49	-96.61	26.63
7	Red Snapper	HB-5-8	23.00	10.84	-18.76	14.10	45.35	-96.77	26.90
8	Red Snapper	HB-5-9	22.00	10.96	-18.51	14.53	45.00	-96.77	26.90
9	G. Amberjack	HB-6-10	33.00	13.94	-17.35	14.50	45.19	-96.60	26.63
10	Red Snapper	HB-6-11	24.00	11.78	-18.16	14.29	44.28	-96.60	26.63
11	G. Amberjack	HB-7-12	33.00	14.70	-16.98	14.24	45.80	-96.61	26.68
12	Red Snapper	CL-1-13	13.50	13.29	-18.00	13.85	45.31	-96.86	26.19
13	Red Snapper	CL-1-14	17.00	13.73	-17.70	14.51	45.30	-96.86	26.19
14	Red Snapper	CL-1-16	18.00	13.56	-18.02	13.93	44.44	-96.86	26.19
15	Red Snapper	CL-1-17	21.75	14.15	-17.56	14.33	44.58	-96.86	26.19
16	Red Snapper	CL-1-18	18.00	13.73	-17.87	13.99	45.56	-96.86	26.19
17	Red Snapper	CL-1-19	12.50	13.55	-17.70	14.31	45.81	-96.86	26.19
18	Red Snapper	CL-1-20	18.00	13.81	-17.94	13.73	45.31	-96.86	26.19
19	Red Snapper	CL-1-21	13.00	13.84	-17.50	14.42	44.93	-96.86	26.19
20	Red Snapper	CL-1-22	17.00	13.46	-17.78	13.98	43.99	-96.86	26.19

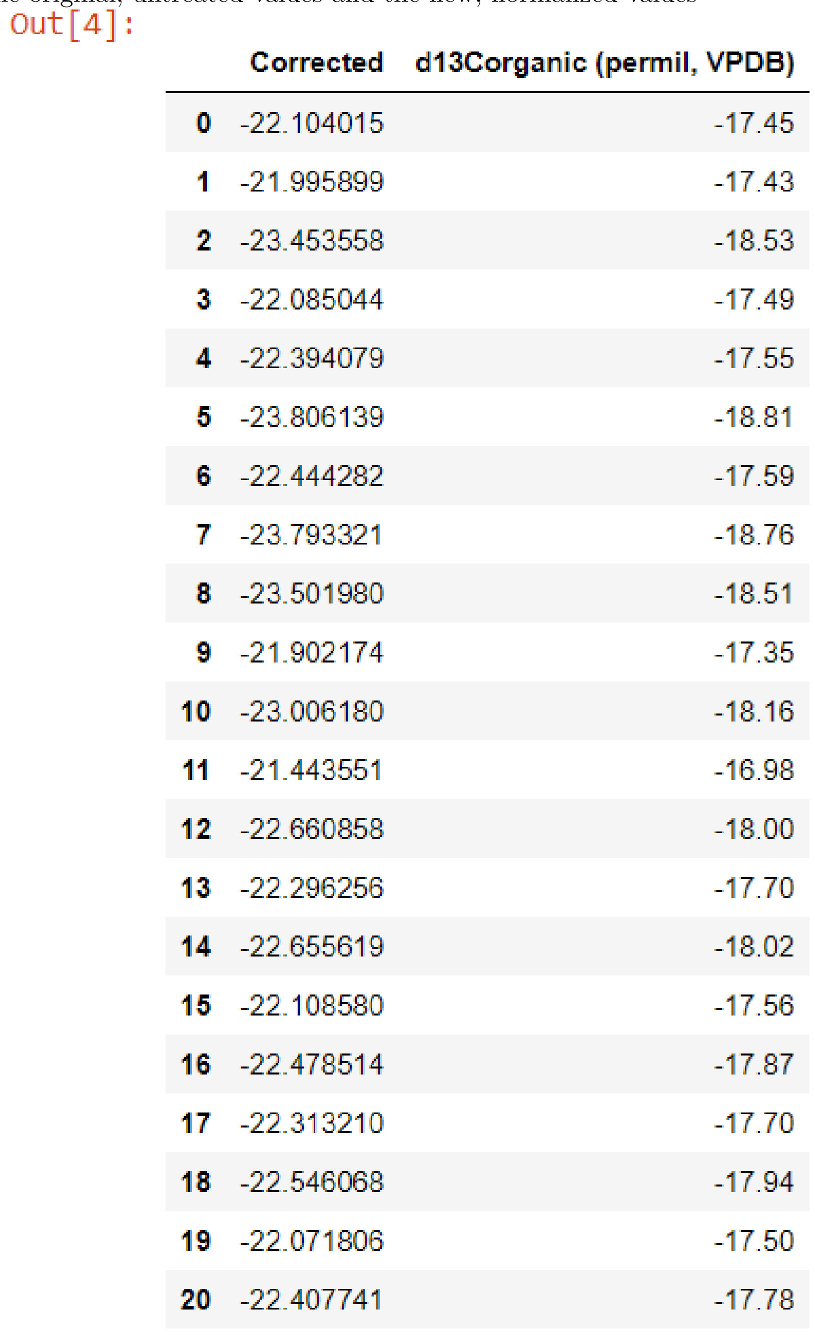
than computing the formula utilizing a ”for-loop”, the choice was made because Pandas could directly interact with the variables containing the needed data and directly apply the formula to them ,making the code more simple, a ”for-loop” would have required that the user imputed each data point manually and store it into an array. This method is time consuming and is not very user-friendly, which defeats one of the purposes of the program. Finally the program stored the corrected values into a new column utilizing Pandas and showcased the updated version of the table (Table 1).

User-friendliness was the other purpose of this program, so it was in need of an easy-to-use graphical user interface (GUI). There are several libraries that are able to create a GUI, however the one chosen for this program was 'ipywidgets' because it is made to create widgets that can be used in Jupyter to create objects that the user can interact and modify aspects of the program. The GUI was built to be able to specify the coordinate limits of a map and show a scatter plot in the map that represents the sampling sites in which the tissues were taken for the isotope analysis. To build a map first a function was made with the help of the Cartopy library, which is a package designed for geospatial data processing. The function added features to the map like land, ocean and coastlines and it also established a way to zoom into the area of the world that most fitted to the user's needs. Then to start building the GUI, four sliders where built and stored into their own variable, the sliders control the values that define the borders of the constructed map, each slider for the x-axis represent any value from -180 to 180 and the sliders of the y-axis controlled numbers from -90 to 90, representing latitude and longitude values. After this the sliders are stored in a new function called "map it" which has the elements necessary for the construction of the map. This new function is called back with another function which activates the "map it" function by clicking a button generated utilizing ipywidgets. Finally, with the help of the "tab" widget the defined functions are stored into a virtual box in a tab. A second tab shows a box plot of both, uncorrected vs corrected values for comparison.

4 Results

The program returns the results and stores them in a new table (Table 2). The program does compile and creates a nice and understandable GUI where the user can input their coordinate limits and it returns a simple map with the sample sites previously inputted in the first tab, while if the user clicks in the second tab it can observe a box plot that graphically showcases the differences between the raw and the corrected values.

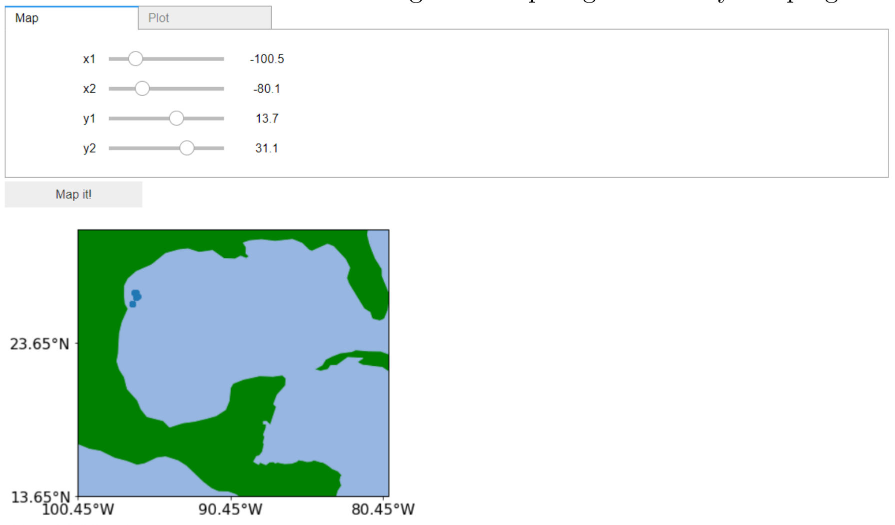
Figure 2: Comparison table returned by the program. It shows the values of the original, untreated values and the new, normalized values



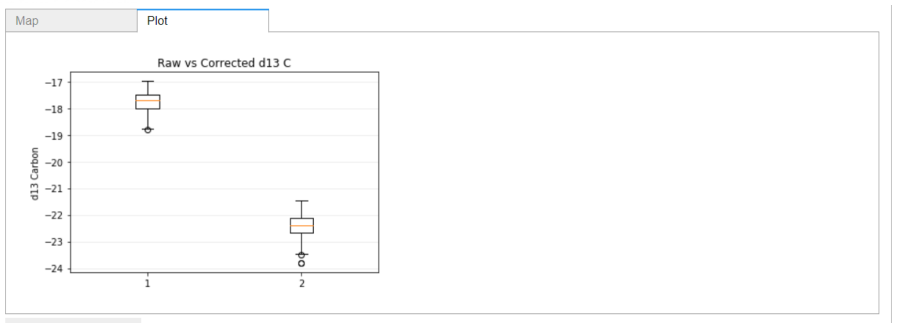
Out[4]:

	Corrected	d13Corganic (permil, VPDB)
0	-22.104015	-17.45
1	-21.995899	-17.43
2	-23.453558	-18.53
3	-22.085044	-17.49
4	-22.394079	-17.55
5	-23.806139	-18.81
6	-22.444282	-17.59
7	-23.793321	-18.76
8	-23.501980	-18.51
9	-21.902174	-17.35
10	-23.006180	-18.16
11	-21.443551	-16.98
12	-22.660858	-18.00
13	-22.296256	-17.70
14	-22.655619	-18.02
15	-22.108580	-17.56
16	-22.478514	-17.87
17	-22.313210	-17.70
18	-22.546068	-17.94
19	-22.071806	-17.50
20	-22.407741	-17.78

Figure 3: Map returned after selecting the desired coordinates and clicking the action button. Second tab showcasing the box-plot generated by the program.



x1 -100.5
x2 -80.1
y1 13.7
y2 31.1



5 Discussion

The program does normalizes the values of Carbon isotopes, however it does have its limitations, for example, if the values are not stored in an excel spreadsheet then the user would not be able to do the correction with this code. Although the program may look simple, however, creating all the functions and fitting them into a GUI was challenging and it took days just to figure out how to do it without having multiple errors. The program can be improved and the interactivity could be improved utilizing a different package like plotly and mapbox, which were strong candidates to be used in this program, the reason they "borrow" their interfaces from a third party and felt like "cheating". Overall the program can be improved but it accomplishes its initial main goal.

References

- Pinnegar, J. K. Polunin, N. V. C. Differential fractionation of 13C and 15N among fish tissues: implications for the study of trophic interactions. 7
- Post, D. M. et al. Getting to the fat of the matter: models, methods and assumptions for dealing with lipids in stable isotope analyses. Oecologia 152, 179–189 (2007).
- Mintenbeck, K., Brey, T., Jacob, U., Knust, R. Struck, U. How to account for the lipid effect on carbon stable-isotope ratio (13 C): sample treatment effects and model bias. J. Fish Biol. 72, 815–830 (2008).