zooaRchGUI User Manual

Erik Otárola-Castillo

2017-05-11

Contents

zooaRch

Zooarchaeology is concerned with the analysis and inference of faunal remains recovered from archaeological sites. The **zooaRch package** provides a *Graphical User Interface (GUI)* to analytical tools for zooarchaeological data in the R package. Functions in this package allow users to:

- input
- handle
- visualize
- and analyze zooarchaeological data

Useful data are available with the package. After starting zooaRch, users will have access to these data. Most datasets are in “wide format”. Included is Binford’s comprehensive forager dataset (Binford 2001) made available by Dr. Amber Johnson from http://ajohnson.sites.truman.edu/data-and-program/. Typing “?LRBdata” into the R (or RStudio) console provides information about the dataset. The key to this dataset is also available and provides users with descriptions of the variable names. The Binford dataset can be used as example for most functions in zooaRch.

Part 1 Installing zooaRch

Install R and RStudio

1. Install R by downloading the latest version from the Cromprehensive R Archive Network (CRAN) https://www.r-project.org/.
2. To make coding easier, you should also download RStudio from https://www.rstudio.com/products/RStudio/#Desktop.

Install the *zooaRch* package

1. FIRST, before installing the zooaRchGUI package, please make sure you have java installed for your Windows (32 and 64 bit) and Apple machines:
   a. https://www.java.com/en/download/manual.jsp
   b. Apple Macintosh machines will also NEED to have XQuartz installed, download from https://www.xquartz.org/
   c. Download and install “JAGS”" (necessary for Bayesian computations) for your particular operating system from the download section in http://mcmc-jags.sourceforge.net/
2. Now that you have these installed, fire up R or RStudio.
3. Install the latest version of zooaRchGUI from
   a. CRAN:

```
install.packages("zooaRchGUI", dependencies=TRUE)
```

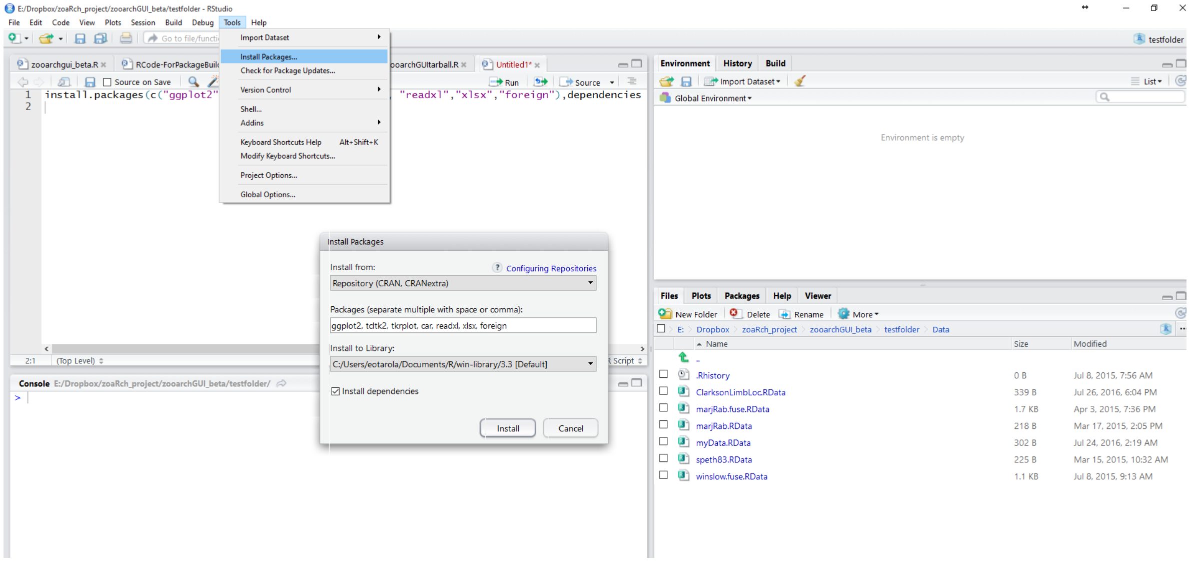
   b. or github (installing devtools()):

```
install.packages("devtools")
library(devtools)
install_github("zooarchGUI/zooaRchGUI", dependencies=TRUE)
```

4. You can install the following necessary packages by copying and pasting the following R code into the RStudio console,

```
install.packages(c("ggplot2","tcltk2", "tkrplot", "car", "readxl","xlsx","foreign",
                   "maptools","spdep","raster", "tripack", "pgirmess", "geomorph",
                   "vegan", "splancs", "rjags", dependencies = TRUE))
```

5. You can also install these packages interactively by typing the names of the packages into RStudio. Get this R-studio dropdown through tools->Install Packages. Make sure "installed dependencies" is checked


E:/Dropbox/zoaRch_project/zooarchGUI_beta/testfolder - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Import Dataset
Install Packages...
Check for Package Updates...
Version Control
Shell...
Addins
Keyboard Shortcuts Help Alt+Shift+K
Modify Keyboard Shortcuts...
Project Options...
Global Options...
zooarchgui_beta.R
RCode-ForPackageBuil
oarchGUItarball.R
Untitled1*
Source on Save
Run
Source
Environment History Build
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
E: Dropbox zoaRch_project zooarchGUI_beta testfolder Data
2:1 (Top Level)
R Script
Console E:/Dropbox/zoaRch_project/zooarchGUI_beta/testfolder/
testfolder

```
install.packages(c("ggplot2"   "readxl","xlsx","foreign"),dependencies
```

Install Packages

Install from: Configuring Repositories

Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma):

ggplot2, tcltk2, tkrplot, car, readxl, xlsx, foreign

Install to Library:

C:/Users/eotarola/Documents/R/win-library/3.3 [Default]

☑ Install dependencies

Install Cancel

Name	Size	Modified
.Rhistory	0 B	Jul 8, 2015, 7:56 AM
ClarksonLimbLoc.RData	339 B	Jul 26, 2016, 6:04 PM
marjRab.fuse.RData	1.7 KB	Apr 3, 2015, 7:36 PM
marjRab.RData	218 B	Mar 17, 2015, 2:05 PM
myData.RData	302 B	Jul 24, 2016, 2:19 AM
speth83.RData	225 B	Mar 15, 2015, 10:32 AM
winslow.fuse.RData	1.1 KB	Jul 8, 2015, 9:13 AM

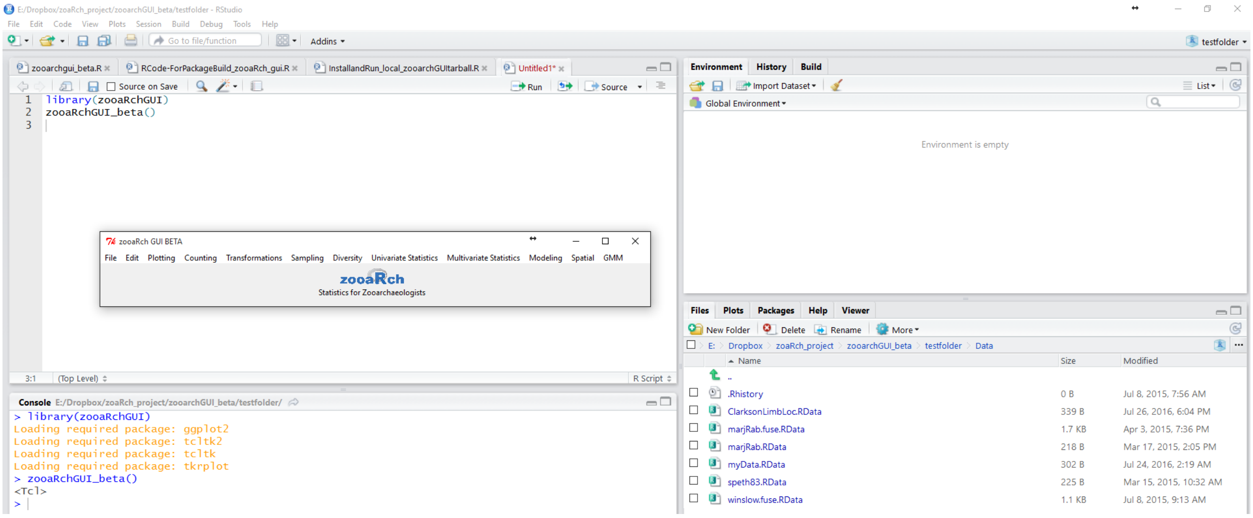
6. Once the package is installed [DONE (zooaRchGUI)], you can load the package by sending the following commands to the R-studio console:

```
library(zooaRchGUI)
```

Start up the *zooaRch GUI* by typing

```
zooaRch_GUI()
```

into the RStudio console. Then, the zooaRch GUI should appear


E:/Dropbox/zoaRch_project/zooarchGUI_beta/testfolder - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
zooarchgui_beta.R
RCode-ForPackageBuild_zooaRch_gui.R
InstallandRun_local_zooarchGUItarball.R
Untitled1*
Source on Save
Run
Source
zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Modeling Spatial GMM
zooaRch
Statistics for Zooarchaeologists
3:1 (Top Level)
R Script
Console E:/Dropbox/zoaRch_project/zooarchGUI_beta/testfolder/
Environment History Build
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
E: Dropbox zoaRch_project zooarchGUI_beta testfolder Data
testfolder

```
library(zooaRchGUI)
zooaRchGUI_beta()
```

```
> library(zooaRchGUI)
Loading required package: ggplot2
Loading required package: tcltk2
Loading required package: tcltk
Loading required package: tkrplot
> zooaRchGUI_beta()
<Tcl>
> 
```

Name	Size	Modified
.Rhistory	0 B	Jul 8, 2015, 7:56 AM
ClarksonLimbLoc.RData	339 B	Jul 26, 2016, 6:04 PM
marjRab.fuse.RData	1.7 KB	Apr 3, 2015, 7:36 PM
marjRab.RData	218 B	Mar 17, 2015, 2:05 PM
myData.RData	302 B	Jul 24, 2016, 2:19 AM
speth83.RData	225 B	Mar 15, 2015, 10:32 AM
winslow.fuse.RData	1.1 KB	Jul 8, 2015, 9:13 AM

That's it! You can browse through the package, import your own data, or use the included datasets in the package (see help file):

Part 2 Using zooaRch

0. Help Menu

0.1 About

The **zooaRch package** provides a *Graphical User Interface (GUI)* to analytical tools for zooarchaeological data in the R computing environment.

0.2 zooaRch User Guide

This will open the user guide to the zooaRchGUI program.

0.3 Example Data

Spatial

Spatial Polygon

This dataset is useful for functions in the spatial menu, particularly with the Ripley's K function. It contains an example of a spatial polygon. It can also be loaded using data(spatPOLY).

Random Spatial Points

This dataset contains an example of completely random points. It is a dataframe that is useful for functions in the spatial menu. It can also be loaded using data(randPTS). These data are particularly useful with the Ripley's K function.

Clustered Spatial Points

This dataset contains an example of random points in clusters. It is a dataframe that is useful for functions in the spatial menu. It can also be loaded using data(clusterPTS).

Multivariate

Binford's Data

This dataset is the forager database from Binford 2001. It can be used with most functions in the program. It can be loaded using data(LRBdata).

Binford's Variable Key

This is a dataframe containing information on all of the variables used in the Binford 2001 dataset. It can be loaded using data(LRBkey).

Site-x-Species

This is an example dataframe. It can be used in most functions in the program. It can be loaded using data(myData).

Chi Squared

Clarkson Deer Data

This is a dataframe containing the anatomical part location at the Clarkson site (Otarola-Castillo 2010 JAS). It is useful in the Chi Squared function. It can be loaded using data(ClarksonLimbLoc).

Survivorship

Garnsey Survival

This is a dataframe containing bison survival data from Speth 1983. It is useful in the survivorship function. It can be loaded using data(speth83).

Marj Rabba Survival

This is a dataframe containing survival data from the Marj Rabba site. It is useful in the survivorship function. It can be loaded using data(marjRab).

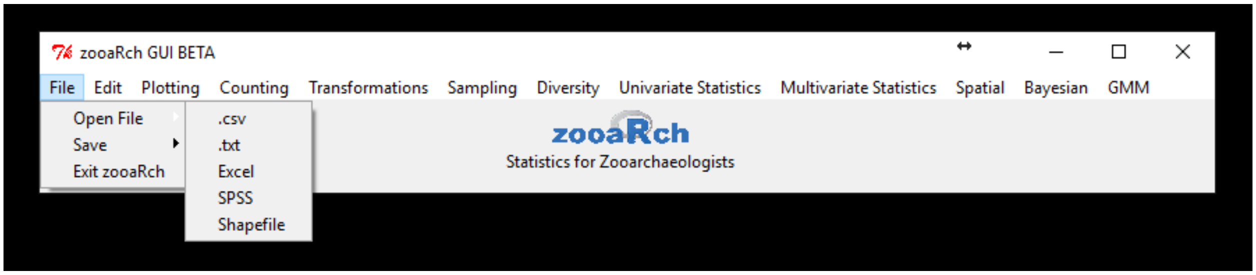
Marj Rabba Fusion

This is a dataframe containing fusion data from the Marj Rabba site. It is useful in the survivorship function. It can be loaded using data(marjRab.fuse).

Winslow Fusion

This is a dataframe containing fusion data from the Winslow site. It is useful in the survivorship function. It can be loaded using data(winslow.fuse).

1. File Menu



1.1 Open File

.csv

This function will open a window that will search a computer directory for .csv files. These files will be saved under the file name as a dataframe.

.txt

This function will open a window that will search a computer directory for .txt files. These files will be saved under the file name as a dataframe.

Excel

This function will open a window that will search a computer directory for Excel files. These files will be saved under the file name as a dataframe.

SPSS

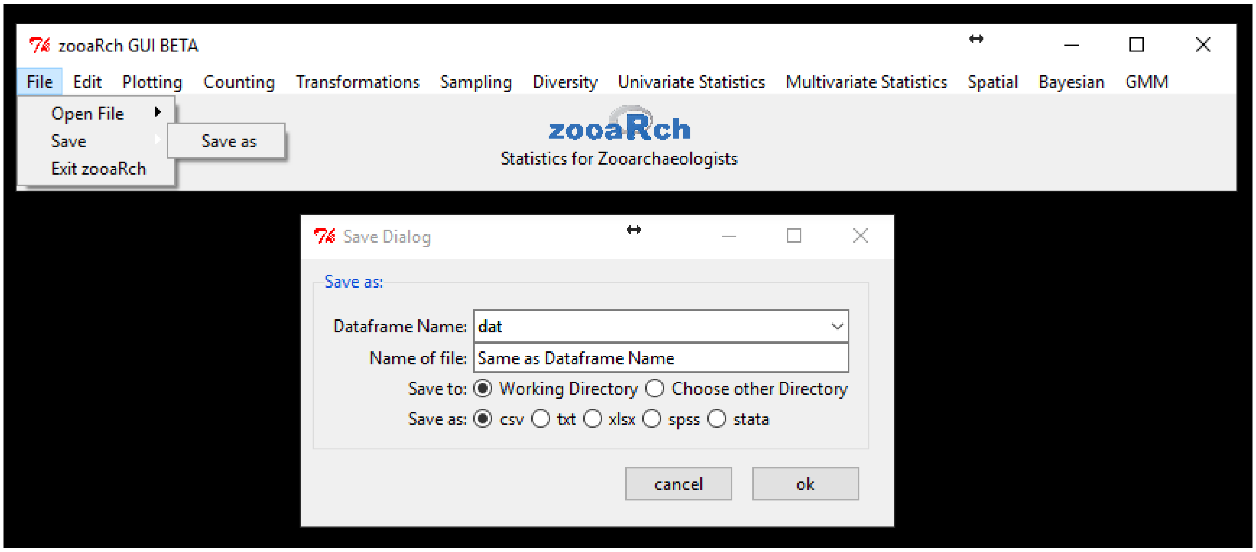
This function will open a window that will search a computer directory for SPSS files. These files will be saved under the file name as a dataframe.

Shapefile

This function will open a window that will search a computer directory for Shapefiles. These files will be saved under the file name as a dataframe.

1.2 Save

Save As



Save as:

Dataframe Name: dat

Name of file: Same as Dataframe Name

Save to: ● Working Directory ○ Choose other Directory

Save as: ● csv ○ txt ○ xlsx ○ spss ○ stata

This function will save inputted dataframes as a file type specified by the user. A dataframe can be saved as a .csv, .txt, .xlsx, .spss, or .stata. The file name can either be the same name as the dataframe or a name given by the user. The file will be saved in either the working directory or a directory given by the user.

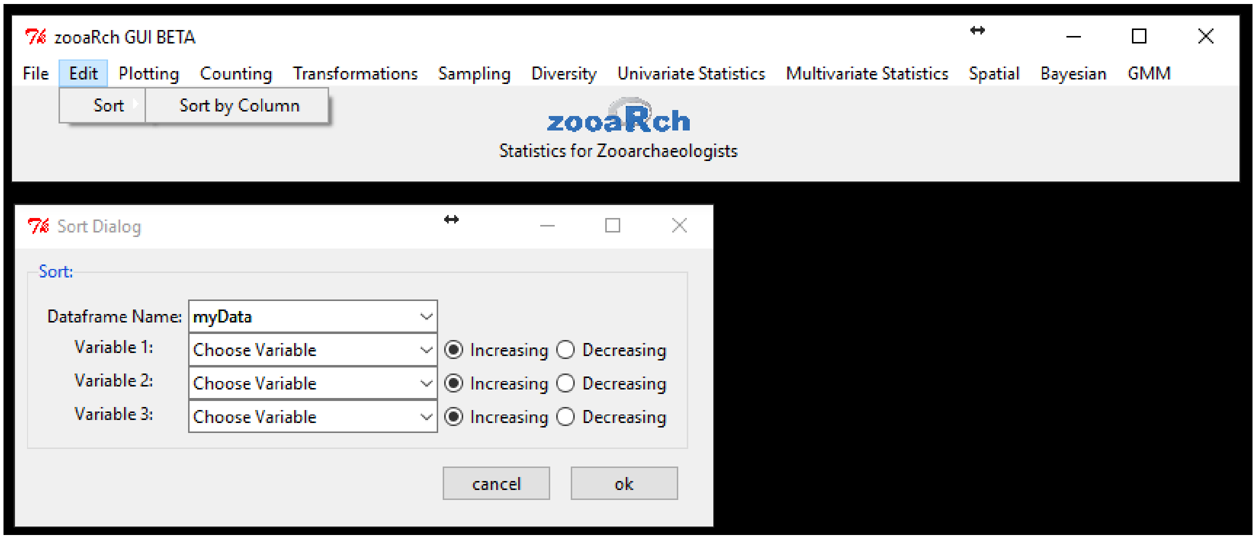
1.3 Exit

This function will close the zooaRch program.

2. Edit Menu

2.1 Sort

Sort by Column


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
Sort
Sort by Column
zooaRch
Statistics for Zooarchaeologists
Sort Dialog
Sort:
cancel
ok

Dataframe Name: myData

Variable 1: Choose Variable ● Increasing ○ Decreasing

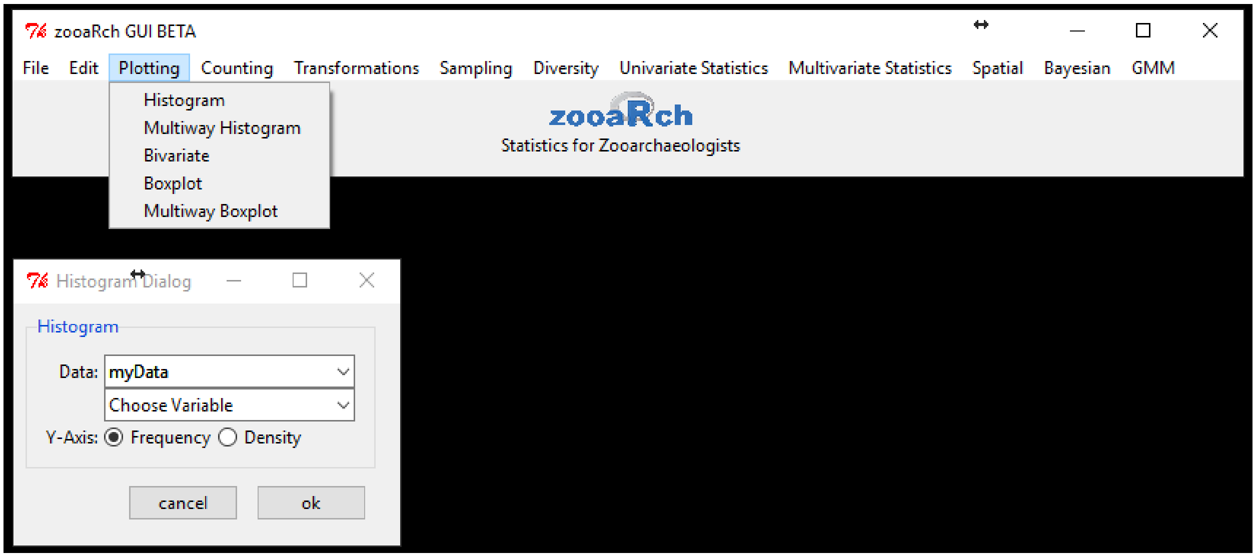
Variable 2: Choose Variable ● Increasing ○ Decreasing

Variable 3: Choose Variable ● Increasing ○ Decreasing

This function will accept a dataframe as the input value. The user can then choose which variables to sort in the dataframe with a maximum number of three variables. The data in the variables can be sorted by increasing or decreasing order. The modified dataframe will be saved in R memory under original filename.

3. Plotting Menu

3.1 Histogram


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
Histogram
Multiway Histogram
Bivariate
Boxplot
Multiway Boxplot
zooaRch
Statistics for Zooarchaeologists
Histogram Dialog
Histogram
cancel
ok

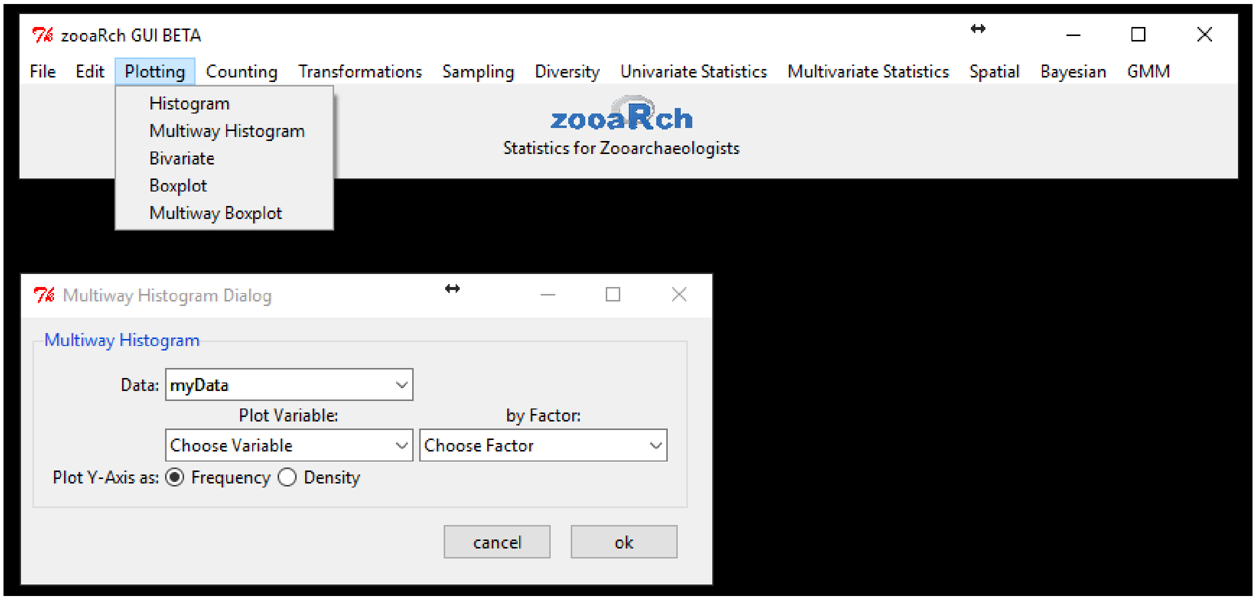
Data: myData

Choose Variable

Y-Axis: ● Frequency ○ Density

This function will create a histogram of the inputted data. It will accept dataframes either already in R memory or loaded by the user. The data used to create a histogram will be selected and saved as a variable. The Y-Axis can be chosen to represent either the frequency or the density of the data represented by the variable. The histogram will be saved in R memory as the variable “Results”.

3.2 Multiway Histogram



Data: myData

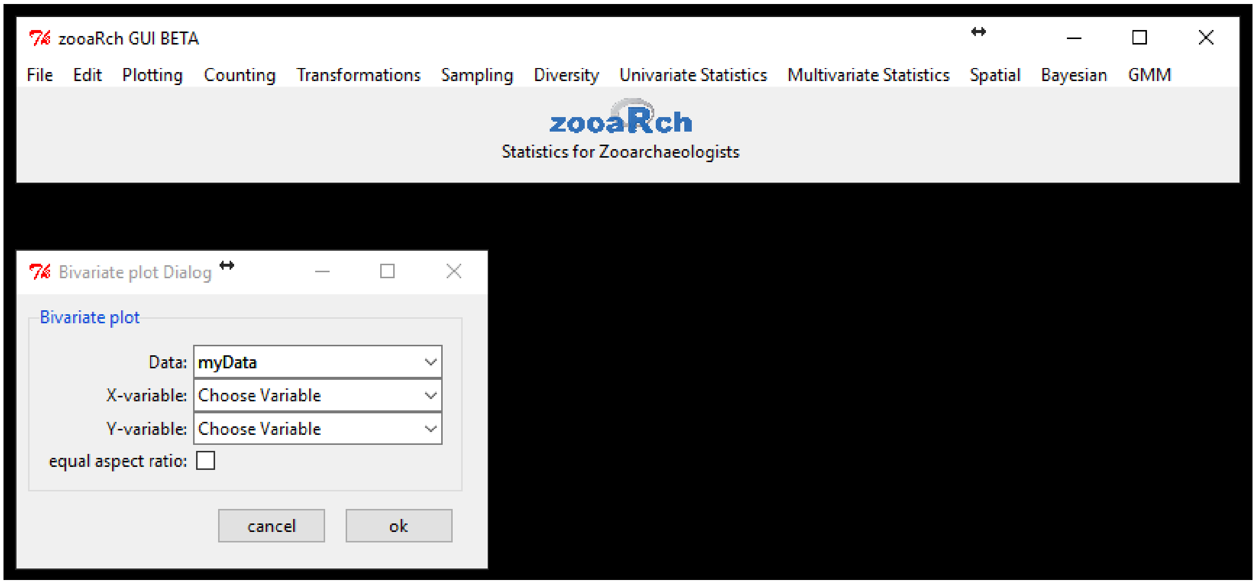
Plot Variable: Choose Variable

by Factor: Choose Factor

Plot Y-Axis as: ● Frequency ○ Density

This function will create a multi-way histogram. It will accept dataframes either already in R memory or loaded by the user. The data used to create the multi-way histogram will be selected and saved as the plot variable. The histogram will be changed by the factor selected from the dataframe in the by factor combobox. The Y-Axis can be chosen to represent either the frequency or the density of the data represented by the variable.

3.3 Bivariate



Data: myData

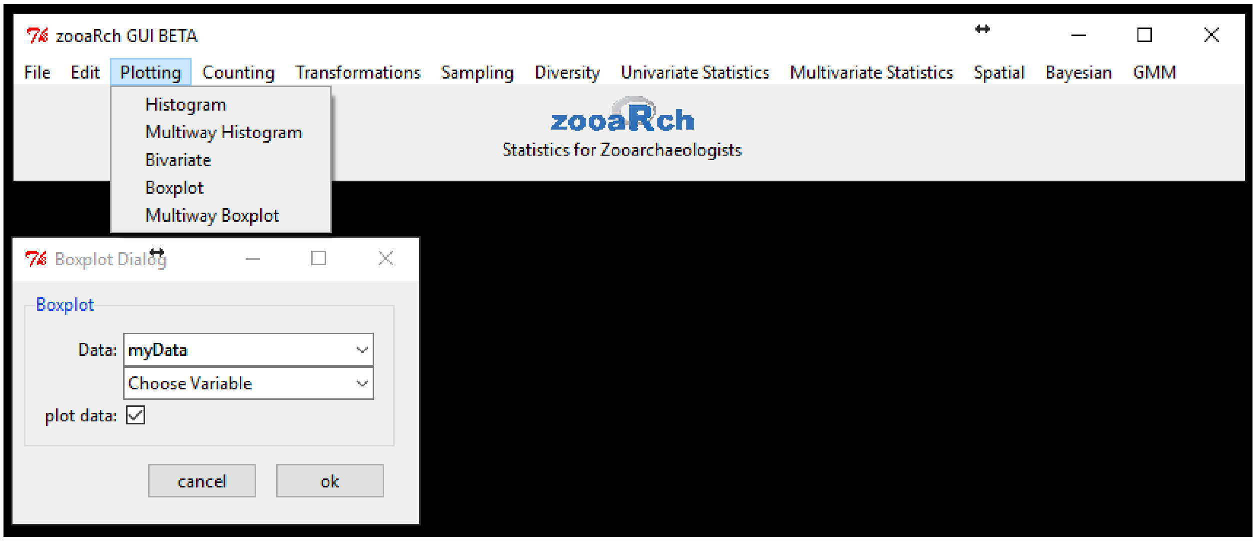
X-variable: Choose Variable

Y-variable: Choose Variable

equal aspect ratio: □

This function will create a bivariate plot. It will accept dataframes either already in R memory or loaded by the user. The first variable in the plot will be selected in the X-variable combobox. The second variable in the plot will be selected in the Y-variable combobox. The plot will have an equal aspect ratio if the checkbox is marked.

3.4 Boxplot



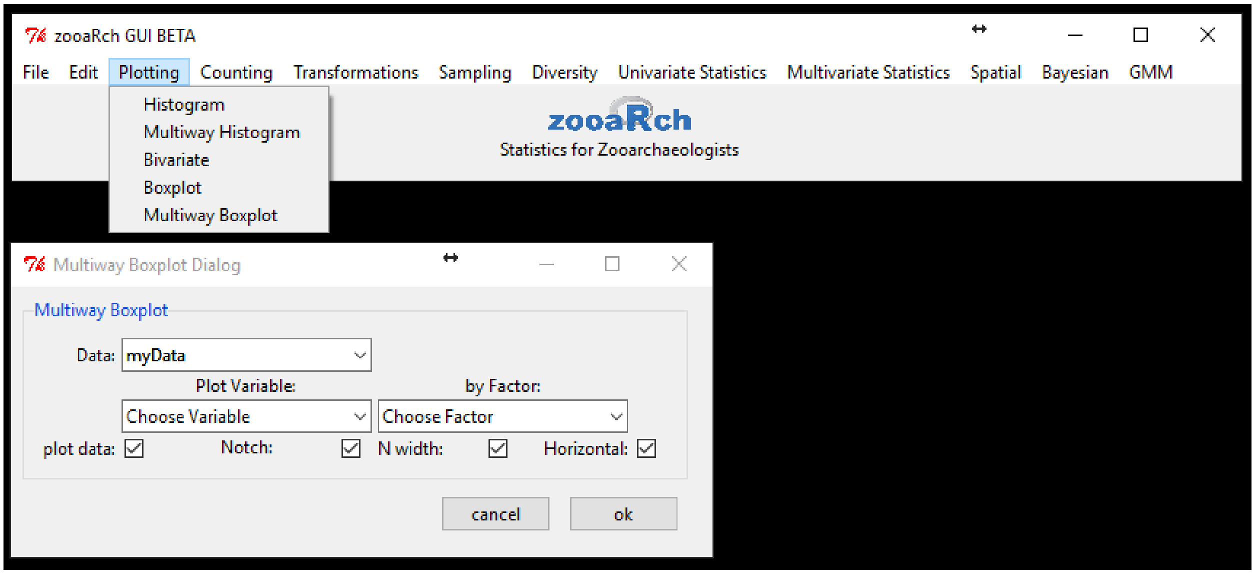
Data: myData

Choose Variable

plot data: ☑

This function will create a boxplot. It will accept dataframes either already in R memory or loaded by the user. The data used to create the boxplot will be selected and saved as a variable. The data will be plotted if the checkbox is marked. The boxplot will be saved in R memory as the variable “Results”.

3.5 Multiway Boxplot



Data: myData

Plot Variable: Choose Variable

by Factor: Choose Factor

plot data: ☑ Notch: ☑ N width: ☑ Horizontal: ☑

This function will create a multi-way boxplot. It will accept dataframes either already in R memory or loaded by the user. The variable to be plotted will be represented by the plot variable. The factor that the boxplot will be modified by is represented by the data in the factor combobox. The data will be plotted if the “plot data” checkbox is marked. The notch will be plotted if the “Notch” checkbox is marked. The width of the boxplot will be proportional to the square-roots of the number of observations in the group if the “N width checkbox is marked. The boxplot will be horizontal if the checkbox”Horizontal" is marked, otherwise the boxplot will be vertical.

4. Counting Menu

4.1 Tabulation

MNI

This function is currently unavailable.

MNE

This function is currently unavailable.

MAU

This function is currently unavailable.

%MAU

This function is currently unavailable.

4.2 LP-Index

LP

This function is currently unavailable.

LP2

This function is currently unavailable.

4.3 ABCML

ABCML

This function is currently unavailable.

4.4 Sexing

Regression

This function is currently unavailable.

DFA

This function is currently unavailable.

Kernal Density

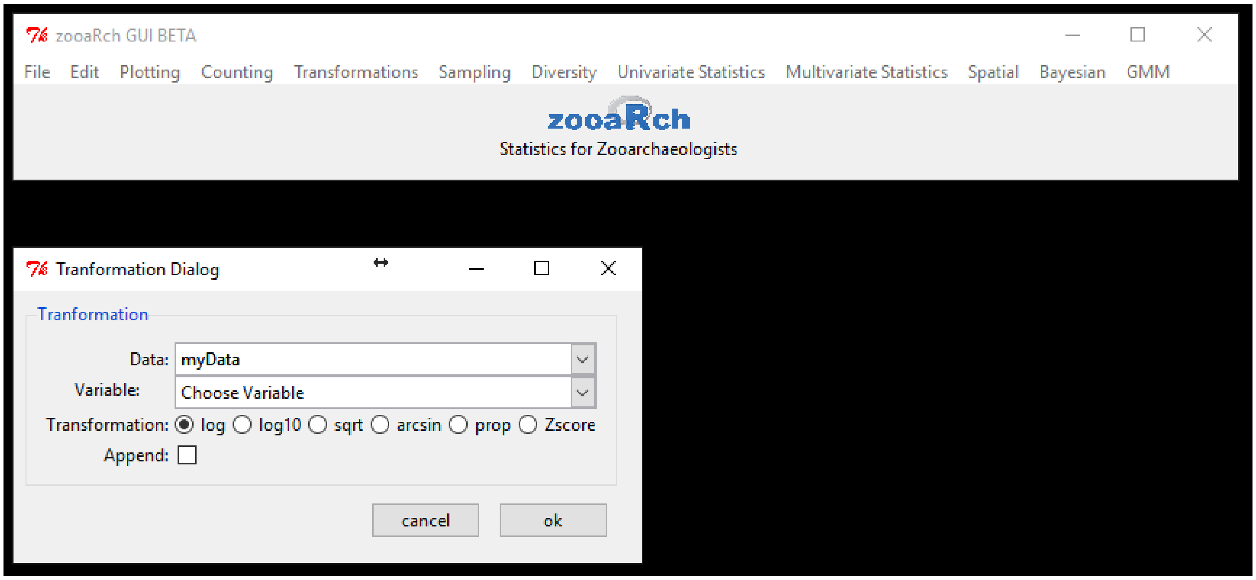
This function is currently unavailable.

Mixing-Model

This function is currently unavailable.

5. Transformations Menu

5.1 Mathematical


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
zooaRch
Statistics for Zooarchaeologists
Tranformation Dialog
Tranformation
cancel
ok

Data: myData

Variable: Choose Variable

Transformation: ● log ○ log10 ○ sqrt ○ arcsin ○ prop ○ Zscore

Append: □

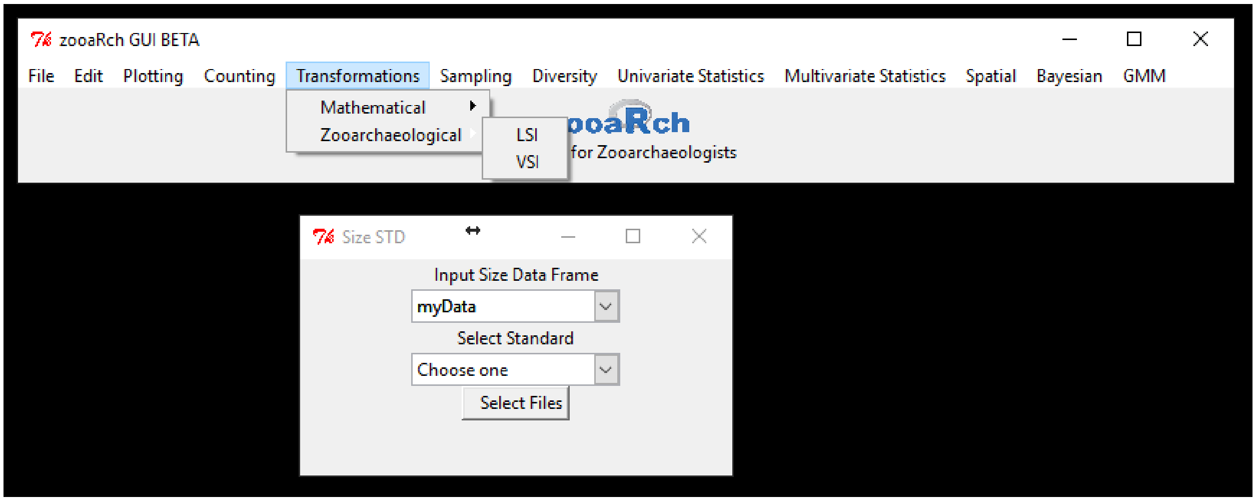
This function will create a mathematical transformation of the inputted data. It will accept dataframes either already in R memory or loaded by the user. The data will be modified using log, log10, sqrt, arcsin, prop, or Zscore depending on which of these options are selected. Results will be saved in R memory under the filename "Results". If the "append" checkbox is marked the output will be added onto any existing "Results" files in R memory.

5.2 Zooarchaeological

LSI

This function will standardize the size of the dataframe. It will accept dataframes either already in R memory or loaded by the user. The column that the dataframe will be standardized to is selected in the "Select Standard" combobox. The resulting data is saved as "standards" in R memory.

VSI


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
Mathematical
Zooarchaeological
LSI
VSI
zooaRch
for Zooarchaeologists
Size STD
Select Files

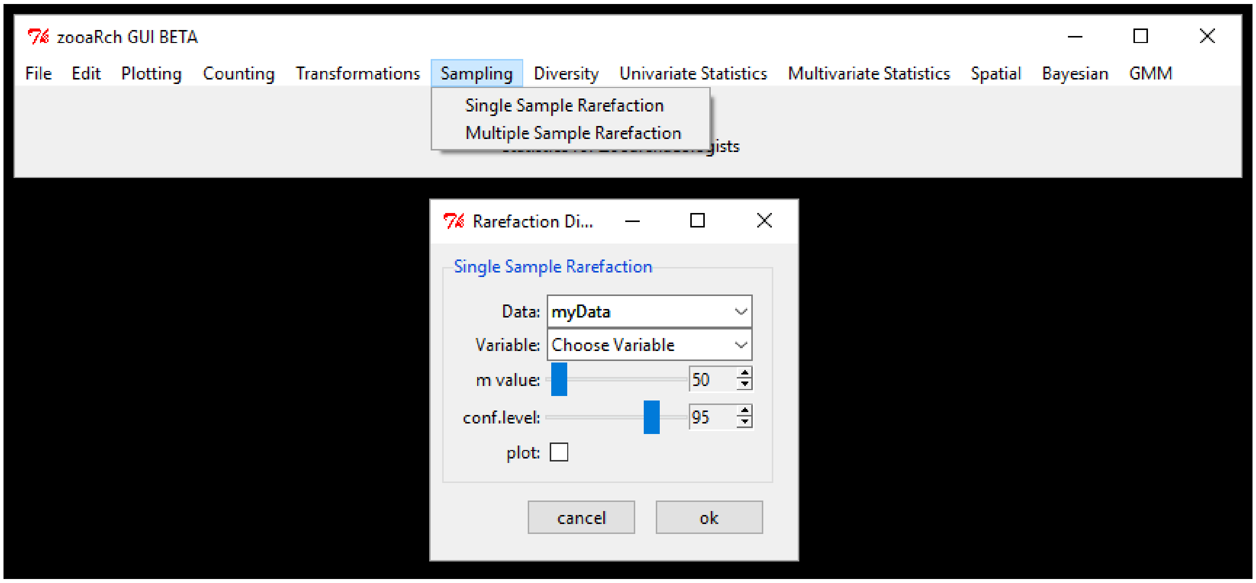
Input Size Data Frame: myData

Select Standard: Choose one

This function will standardize the size of the dataframe. It will accept dataframes either already in R memory or loaded by the user. The column that the dataframe will be standardized to is selected in the “Select Standard” combobox. The resulting data is saved as “standards” in R memory.

6. Sampling Menu

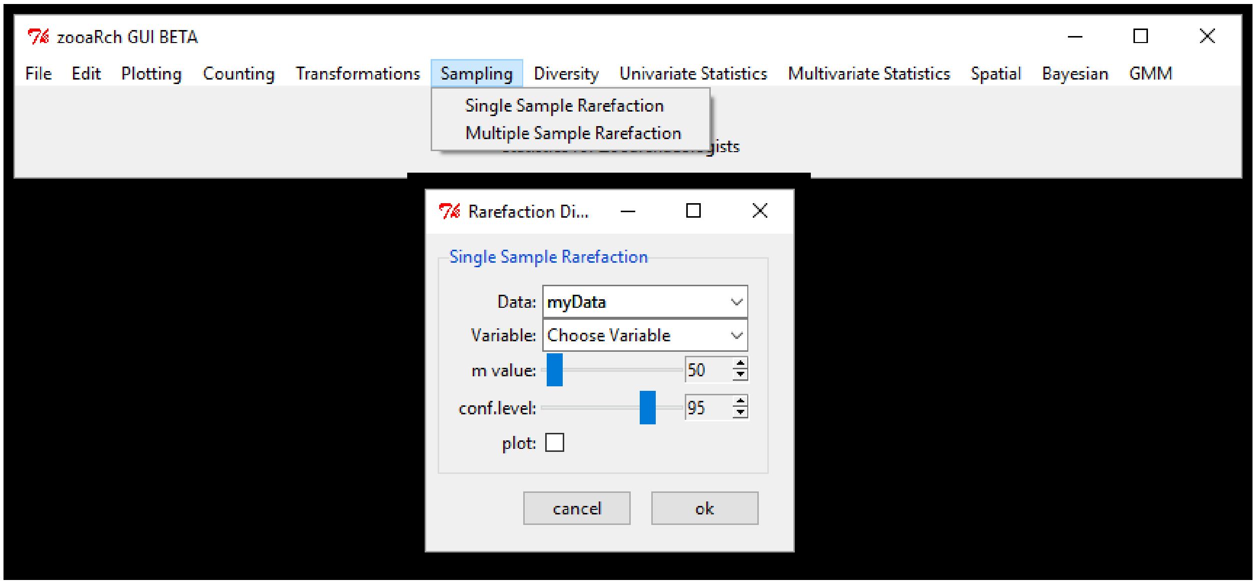
6.1 Single Sample Rarefaction



Data: myData
Variable: Choose Variable
m value: 50
conf.level: 95
plot: □

This function will perform a rarefaction analysis of a single sample. It will accept dataframes either already in R memory or loaded by the user. The variable that the analysis will be done on is chosen along with the margins in which the index is computed and the conf.levels. The results can be plotted if the “plot” checkbox is marked. The results of the analysis are saved in R memory as “Results”.

6.2 Multiple Sample Rarefaction*



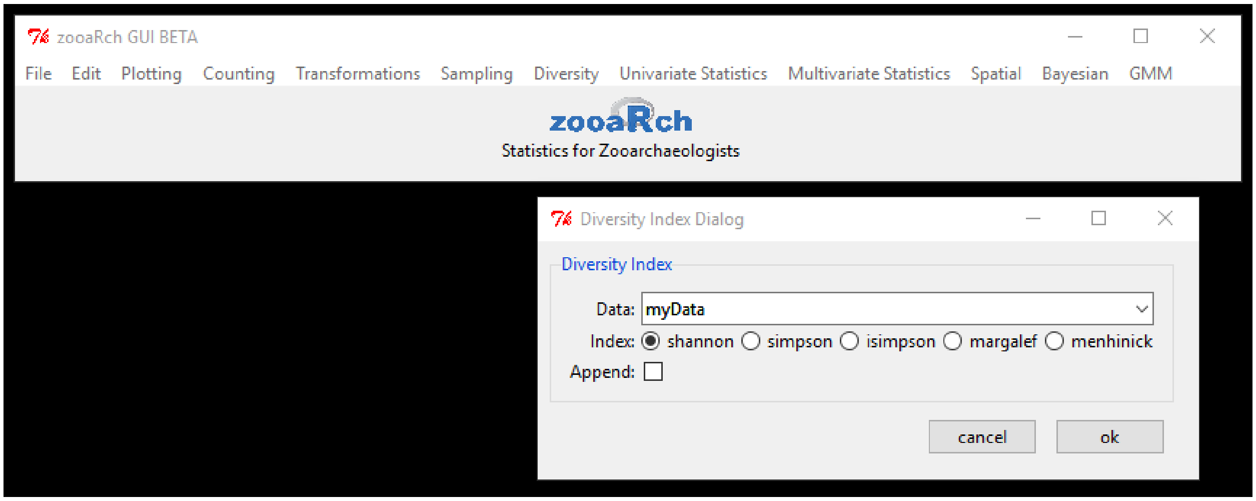
Data: myData
Variable: Choose Variable
m value: 50
conf.level: 95
plot: □

This function will perform a rarefaction analysis for multiple samples. It will accept dataframes either already in R memory or loaded by the user. The variable that the analysis will be done on is chosen along with the margins in which the index is computed and the conf.levels. The results can be plotted if the “plot” checkbox is marked. The results of the analysis are saved in R memory as “Results”.

7. Diversity Menu

7.1 Diversity

Indices



Data: myData

Index: ● shannon ○ simpson ○ isimpson ○ margalef ○ menhinick

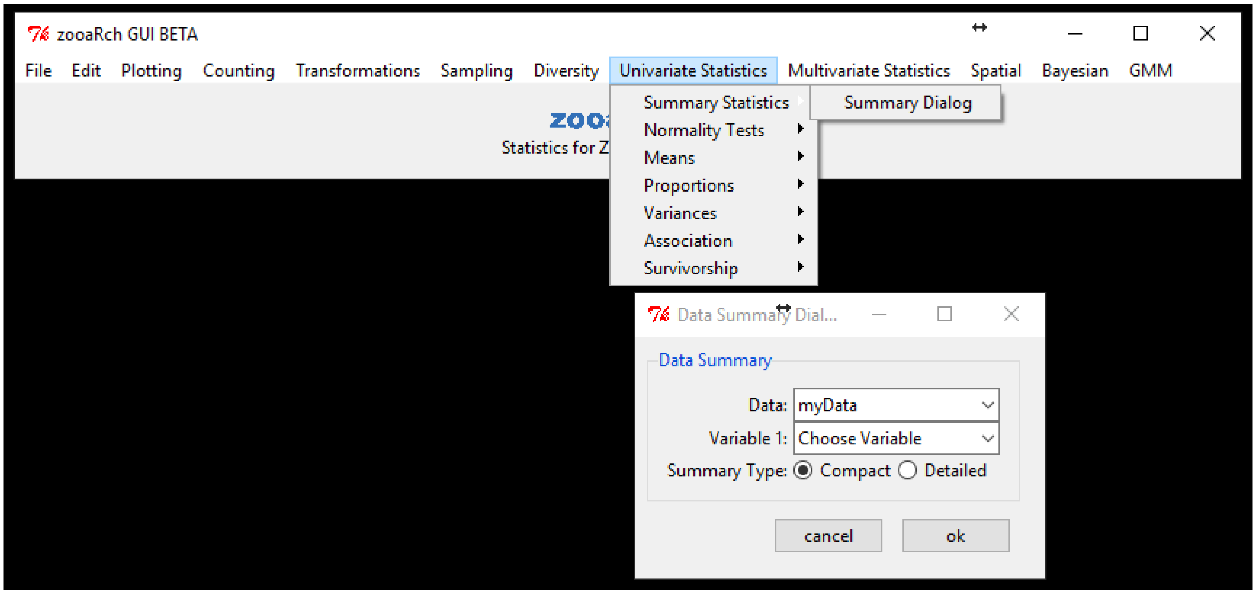
Append: □

This function will create the indices of a dataset. It will accept dataframes either already in R memory or loaded by the user. The indices that can be chosen are: shannon, simpson, isimpson, margalef, menhinick. The results can be appended to the original dataset if the “append” checkbox is marked, or saved in R memory as “Results”.

8. Univariate Statistics Menu

8.1 Summary Statistics

Summary Dialog



Data: myData

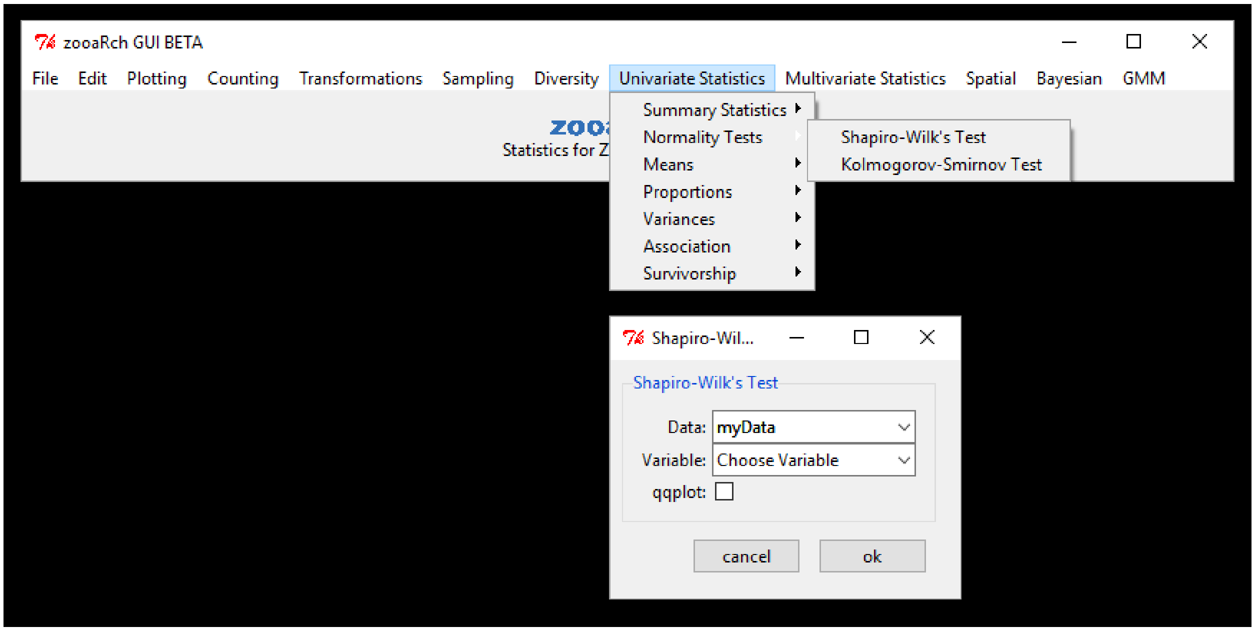
Variable 1: Choose Variable

Summary Type: ● Compact ○ Detailed

This function will create a summary of the variable selected. It will accept dataframes either already in R memory or loaded by the user. The summary can be compact, or detailed. The outputs include the variance, standard deviation, interquartile range, mean, median, skewness and percentiles of data. The results are printed in the console and saved in R memory as “Results”.

8.2 Normality Test

Shapiro-Wilk’s Test



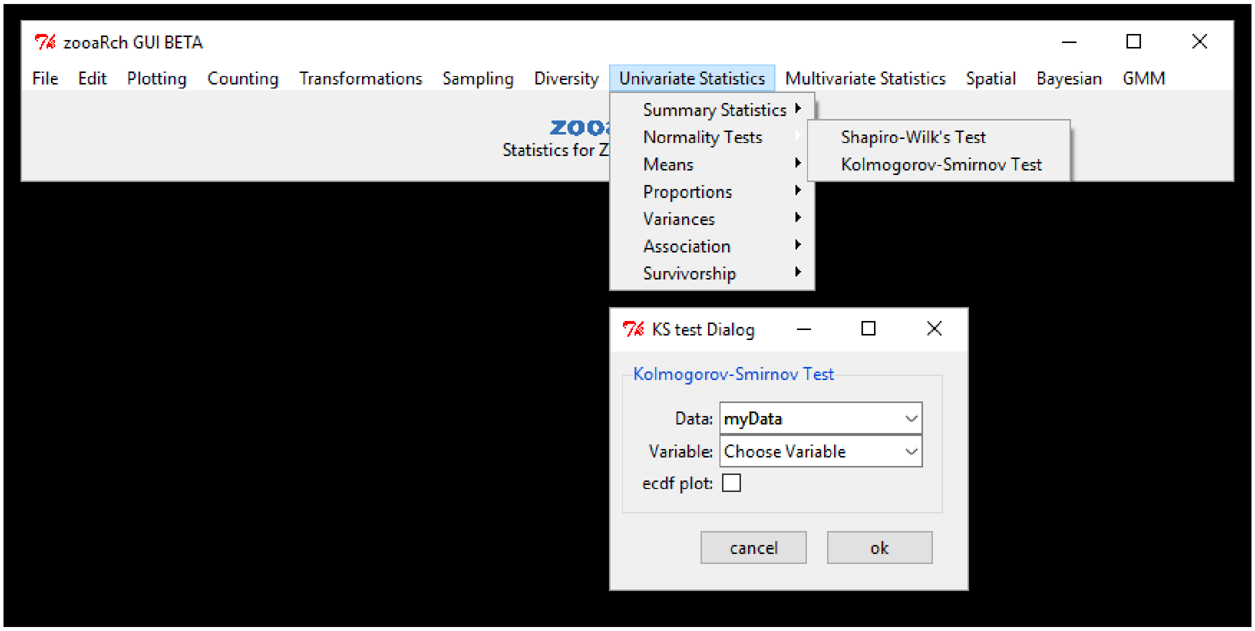
Data: myData

Variable: Choose Variable

qqplot: □

This function will perform a Shapiro-Wilk’s test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The variable from the dataset that the test will be performed on is chosen in the “Variable” combobox. The results will be plotted if the “qqplot” checkbox is marked. The results will be saved in R memory as “Results”.

Kolmogorov-Smironov Test



Kolmogorov-Smirnov Test

Data: myData

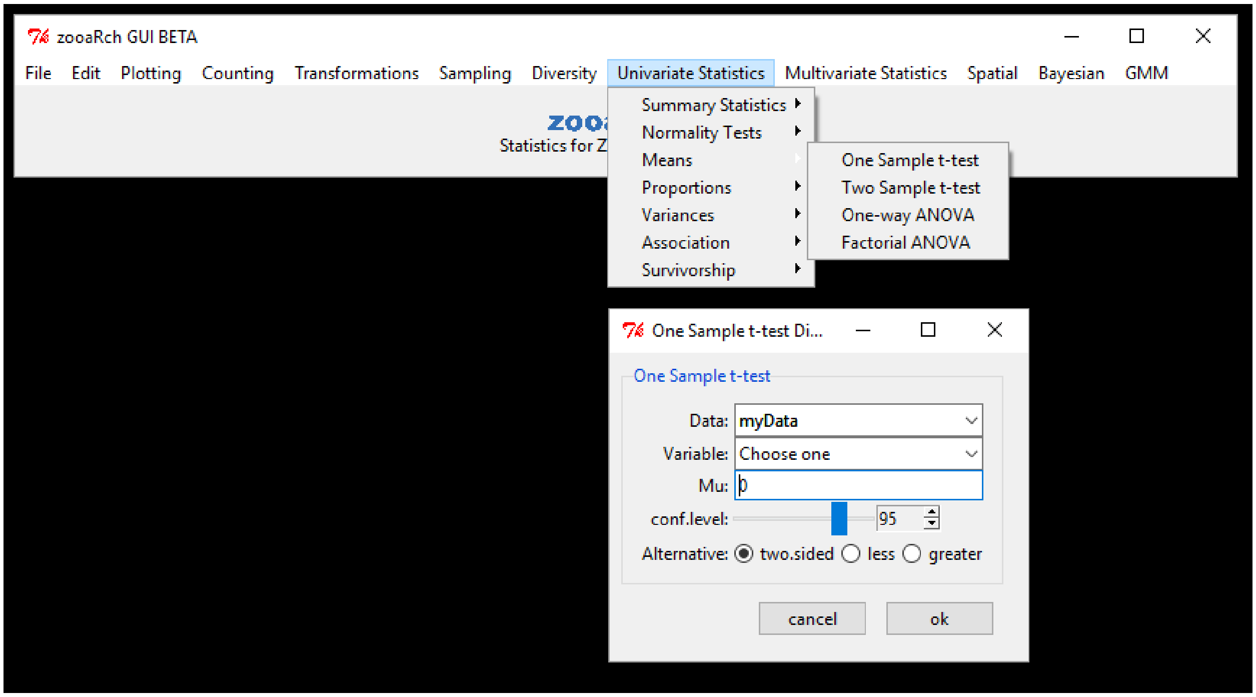
Variable: Choose Variable

ecdf plot: □

This function will perform a Kolmogorov-Smirov test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The variable from the dataset that the test will be performed on is chosen in the "Variable" combobox. An edcf plot will be created if the "edcf plot" checkbox is marked. The results will be saved in R memory as "Results".

8.3 Means

One Sample T-Test



One Sample t-test

Data: myData

Variable: Choose one

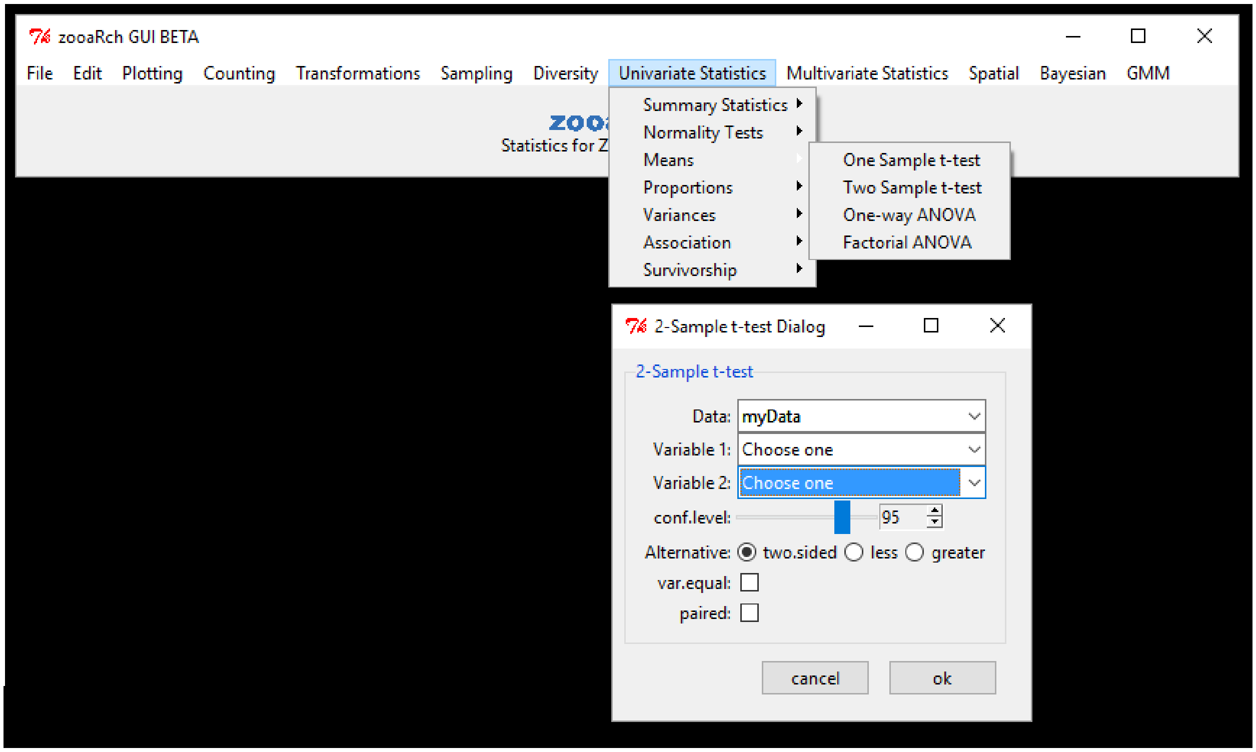
Mu: 0

conf.level: 95

Alternative: ● two.sided ○ less ○ greater

This function will perform a t-test on one sample. It will accept dataframes either already in R memory or loaded by the user. The variable from the dataset is chosen in the “Variable” combobox. The Mu value is inputted by the user along with the conf.level. The Alternative can be chosen to be either two-sided, less than, or greater than. The result are printed into the console and saved in R memory as “Results”.

Two Sample T-Test



Data: myData

Variable 1: Choose one

Variable 2: Choose one

conf.level: 95

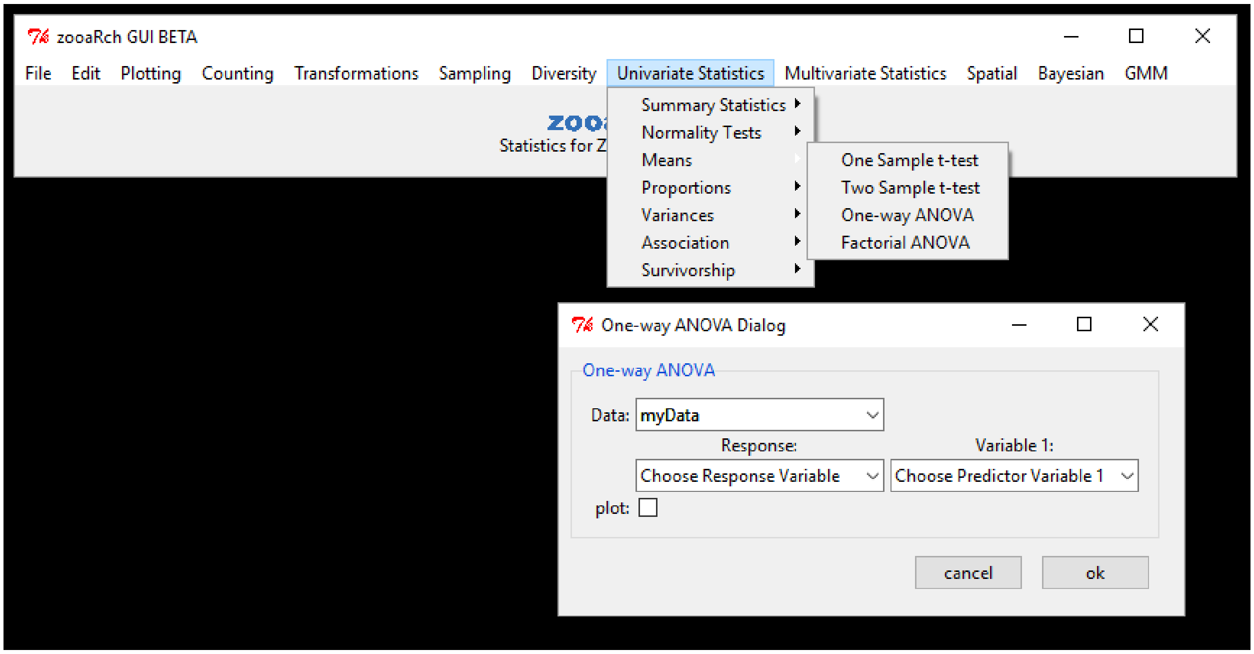
Alternative: ● two.sided ○ less ○ greater

var.equal: □

paired: □

This function will perform a t-test on two samples. It will accept dataframes either already in R memory or loaded by the user. The two variables for the test are chosen in their respective comboboxes. The conf.level is decided with the slider. The alternative can either be two-sided, less than, or greater than. The variables can be equal and/or paired if the checkboxes “var.equal” and “paired” respectively are marked. The results are printed into the consoles and saved in R memory as “Results”.

One-Way ANOVA



One-way ANOVA

Data: myData

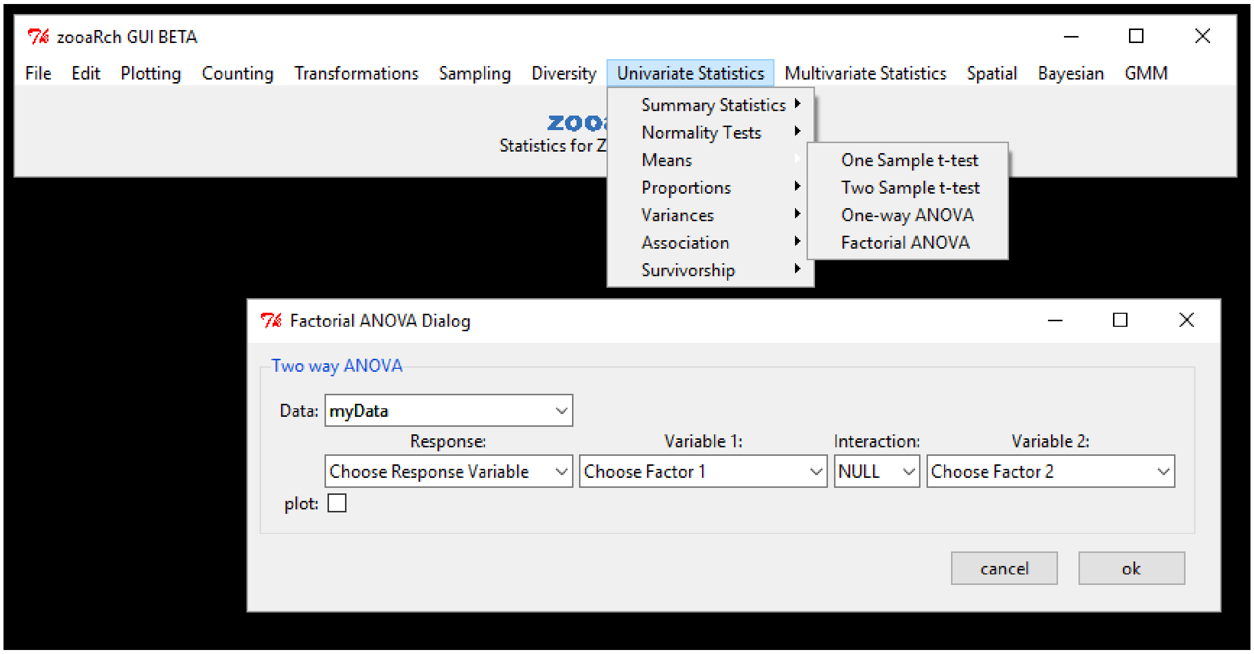
Response: Choose Response Variable

Variable 1: Choose Predictor Variable 1

plot: □

This function will perform a one-way ANOVA test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The response variable is chosen using the "Response" combobox. The predictor variable is selected using the "Variable 1" combobox. The results can be plotted if the "plot" checkbox is marked. The results of the test are saved in R memory as "Results". The results and summary are also printed into the console.

Factorial ANOVA



Two way ANOVA

Data: myData

Response: Choose Response Variable

Variable 1: Choose Factor 1

Interaction: NULL

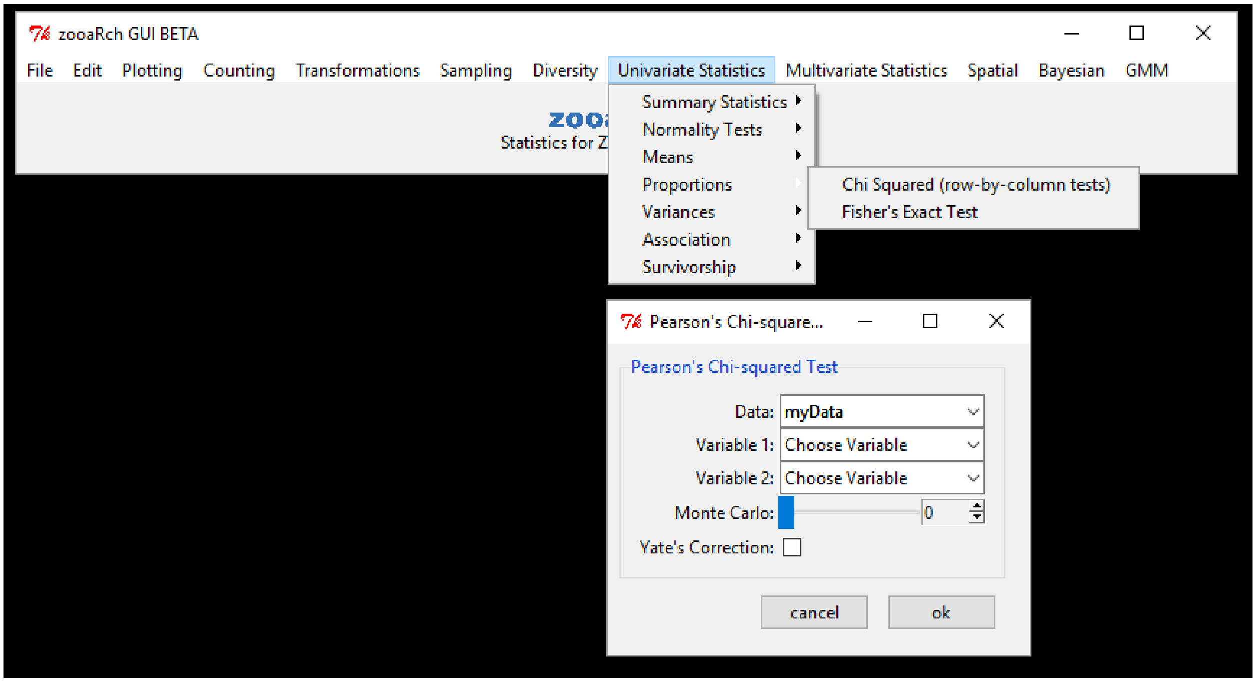
Variable 2: Choose Factor 2

plot: □

This function will perform a factorial ANOVA test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The response variable is chosen using the "Response" combobox. The first and second variables are selected using the "Variable 1" and "Variable 2" respectively. The interaction between the two variables can be either multiplication or addition depending on the input in the "interaction" combobox. The results will be saved in R memory as “Results”. The results and summary are printed to the console. The result can be plotted if the “plot” checkbox is marked.

8.4 Proportions

Chi Squared (row-by-column tests)



Pearson's Chi-squared Test

Data: myData

Variable 1: Choose Variable

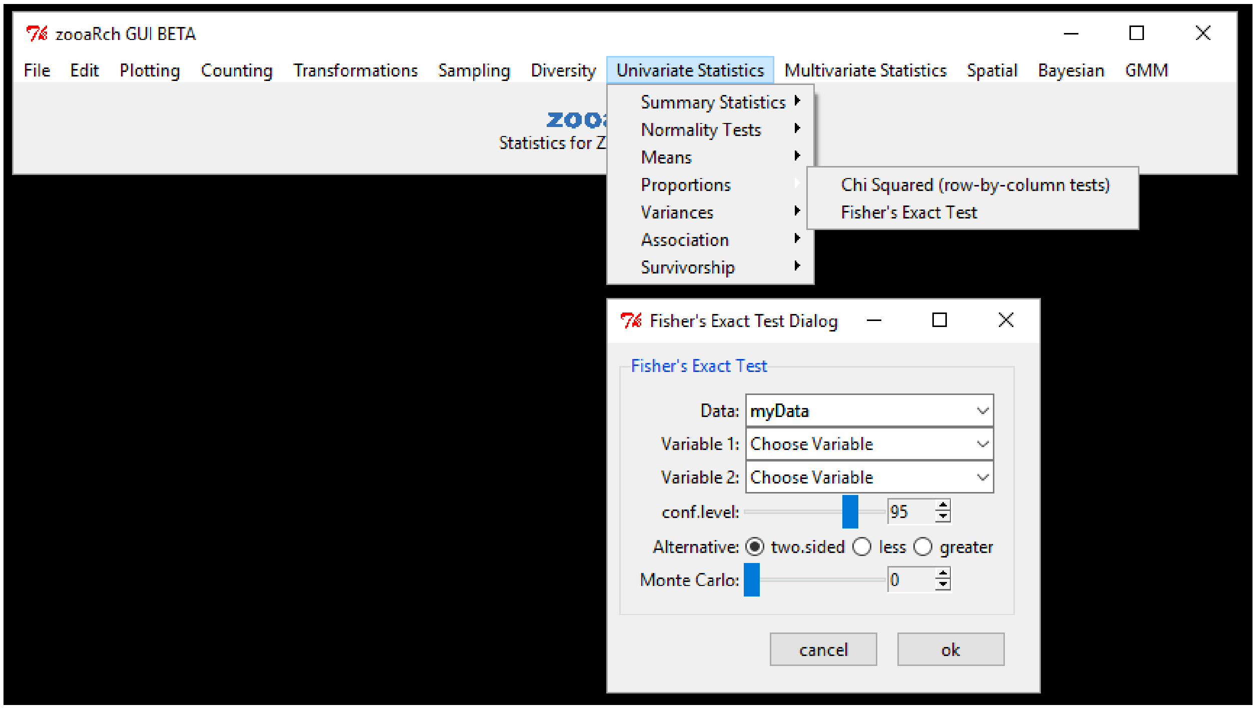
Variable 2: Choose Variable

Monte Carlo: 0

Yate's Correction: □

This function will perform the Chi Squared row-by-column test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The first and second variable for the test *must be categorical* and can be selected in the “Variable 1” and “Variable 2” comboboxes respectively. The Monte Carlo value is selected using the slider, if a permutation test is desired. A Yate’s Correction will be performed during the test if the “Yate’s Correction” is marked. The results are saved to the R memory as “Results”. The results are also printed in the console. Use *data(“ClarksonLimbLoc”)* for an example, or choose appropriate categorical vaiables from *data(LRBdata)*.

Fisher’s Exact Test



Fisher's Exact Test

Data: myData

Variable 1: Choose Variable

Variable 2: Choose Variable

conf.level: 95

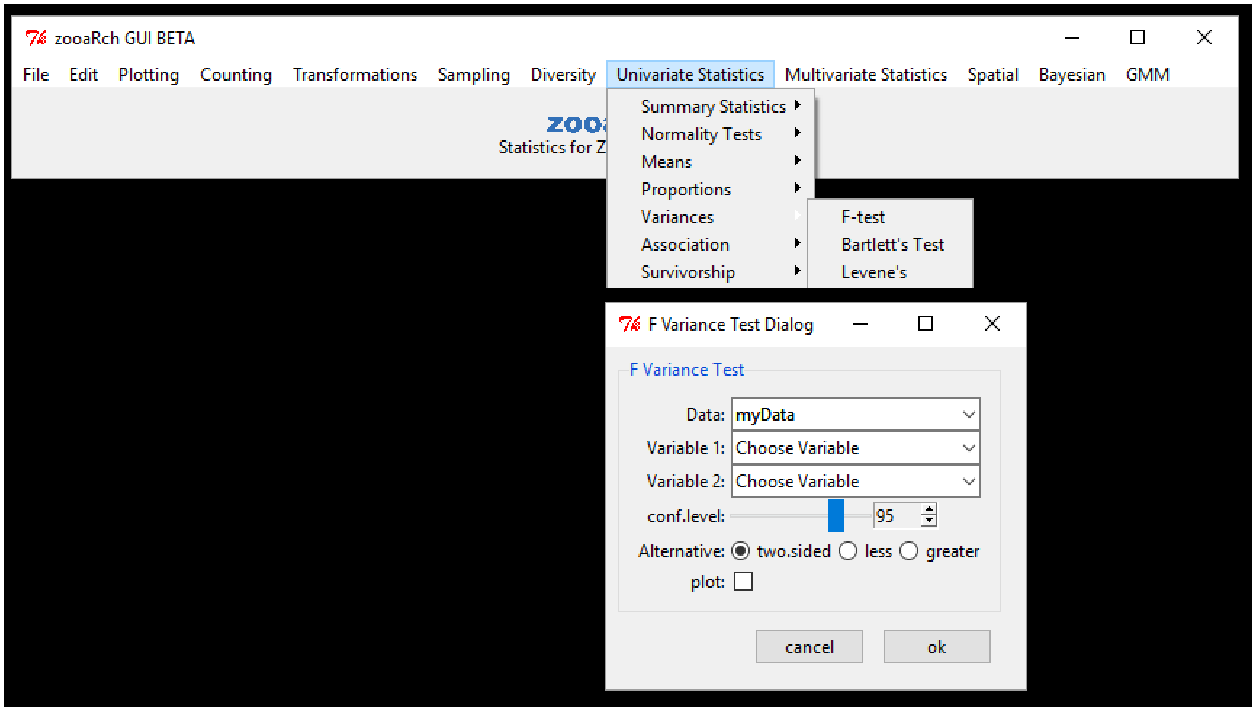
Alternative: ● two.sided ○ less ○ greater

Monte Carlo: 0

This function will perform the Fisher's Exact test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The first and second variable for the test *must be categorical* and can be selected in the "Variable 1" and "Variable 2" comboboxes respectively. The conf.level is chosen using the conf.level slider. The alternative can either be two sided, less than, or greater than. The Monte Carlo value is selected using the Monte Carlo slider. The results of the test will be printed out in the console and plotted. The results will be saved in R memory as "Results". Use *data("ClarksonLimbLoc")* for an example, or choose appropriate categorical vaiables from *data(LRBdata)*.

8.5 Variances

F-Test



F Variance Test

Data: myData

Variable 1: Choose Variable

Variable 2: Choose Variable

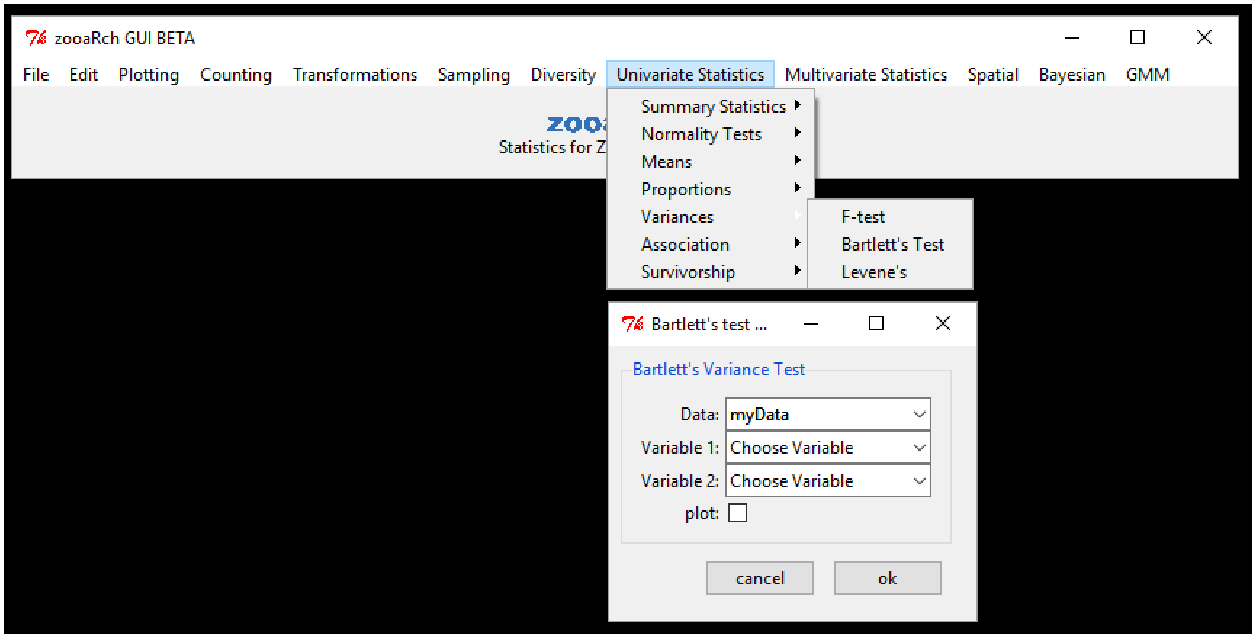
conf.level: 95

Alternative: ● two.sided ○ less ○ greater

plot: □

This function will perform a F Variance test on the selected dataset. The Monte Carlo value is selected using the slider. It will accept dataframes either already in R memory or loaded by the user. The first and second variable for the test will be selected in the “Variable 1” and “Variable 2” comboboxes respectively. The conf.level is chosen using the conf.level slider. The alternative can be either two sided, less than, or greater than. The results can be plotted if the “plot” checkbox is marked. The results are printed in the console and saved in R memory as “Results”.

Bartlett’s Test



Bartlett's Variance Test

Data: myData

Variable 1: Choose Variable

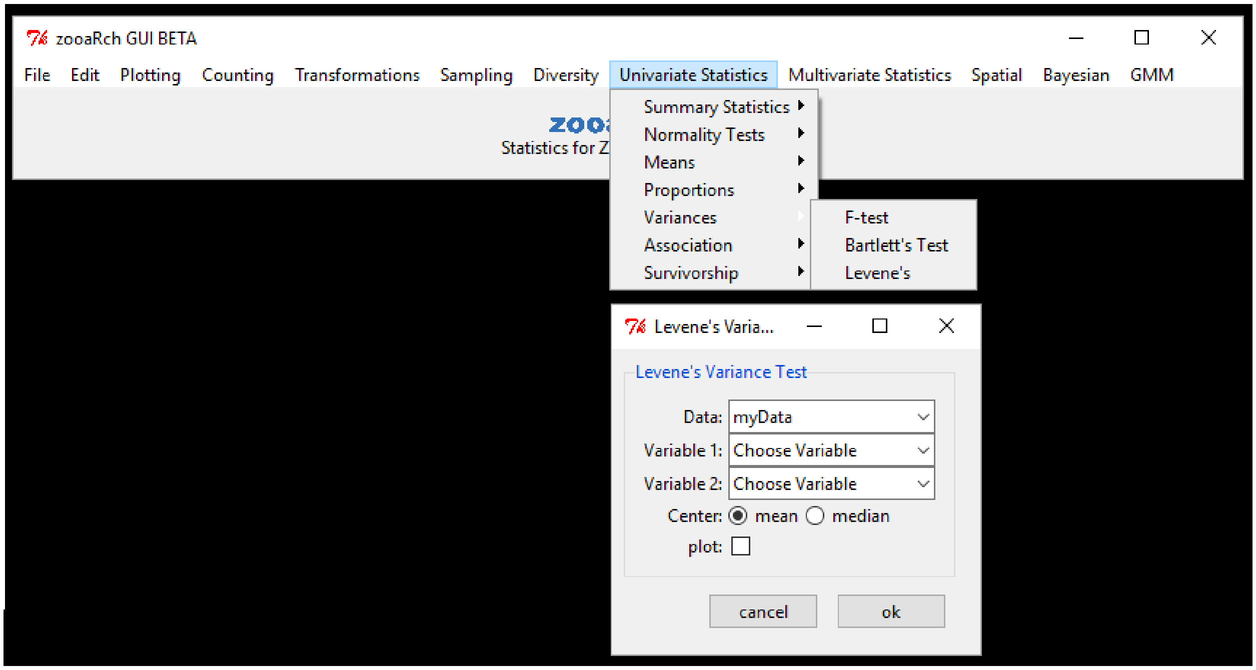
Variable 2: Choose Variable

plot: □

This function will perform a Bartlett’s Variance test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The first and second variables for the test will be selected in the

“Variable 1” and “Variable 2” comboboxes respectively. The results can be plotted if the “plot” checkbox is marked. The results are printed onto the console and saved in R memory as “Results”.

Levene’s



Levene's Variance Test

Data: myData

Variable 1: Choose Variable

Variable 2: Choose Variable

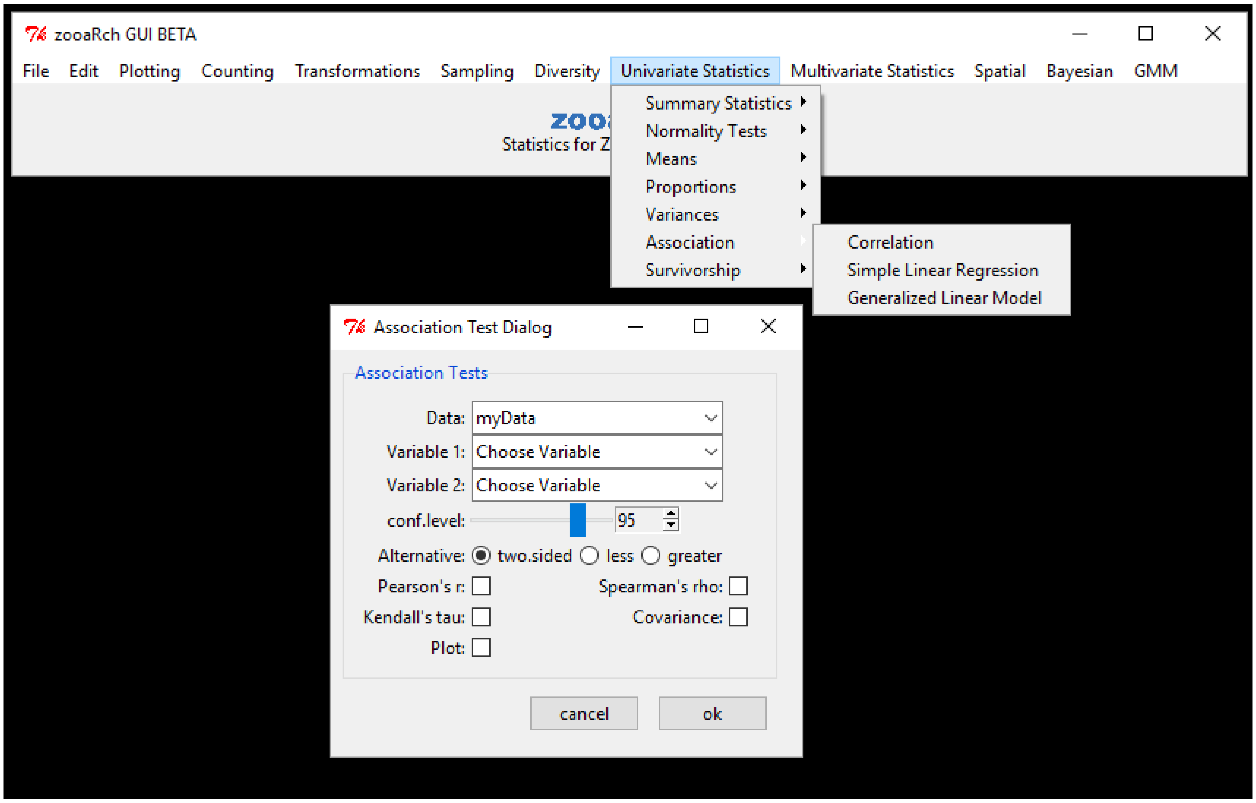
Center: ● mean ○ median

plot: □

This function will perform a Levene’s Variance test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The first and second variables for the test will be selected in the “Variable 1” and “Variable 2” comboboxes respectively. The center can either be the median of the data or the mean of the data. The results can be plotted if the “plot” checkbox is marked. The results are printed in the console and saved in R memory as “Results”.

8.6 Association

Correlation



Data: myData

Variable 1: Choose Variable

Variable 2: Choose Variable

conf.level: 95

Alternative: ● two.sided ○ less ○ greater

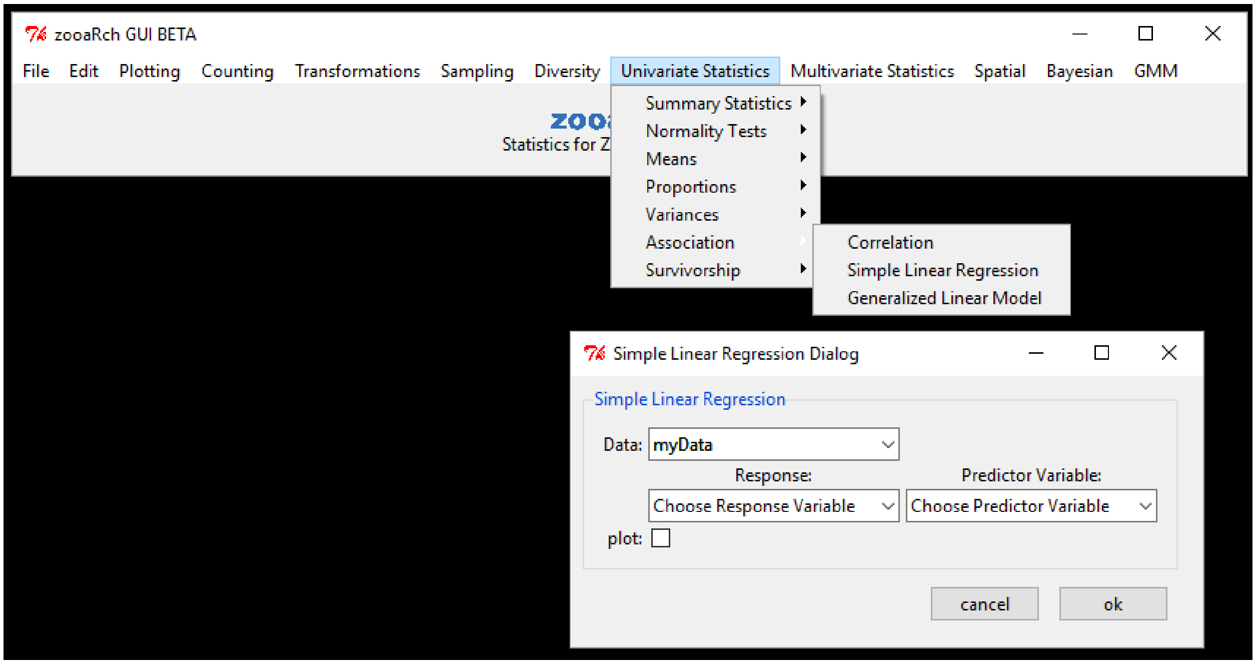
Pearson's r: □ Spearman's rho: □

Kendall's tau: □ Covariance: □

Plot: □

This function will perform an Association test on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The first and second variables for the test will be selected in the "Variable 1" and "Variable 2" comboboxes respectively. The conf.level will be selected using the conf.level slider. The alternative can either be tow sided, less than, or greater than. The Pearson's r, Kendall's tau, Spearman's rho, and the Covariance can be calculated if their respective checkboxes are marked. If calculated they will be saved in R memory as "Pearson_Results", "Kendall_Results", "Spearman_Results", and "Covariance_Results" respectively. The results can be plotted if the "plot" checkbox is marked.

Simple Linear Regression



Data: myData

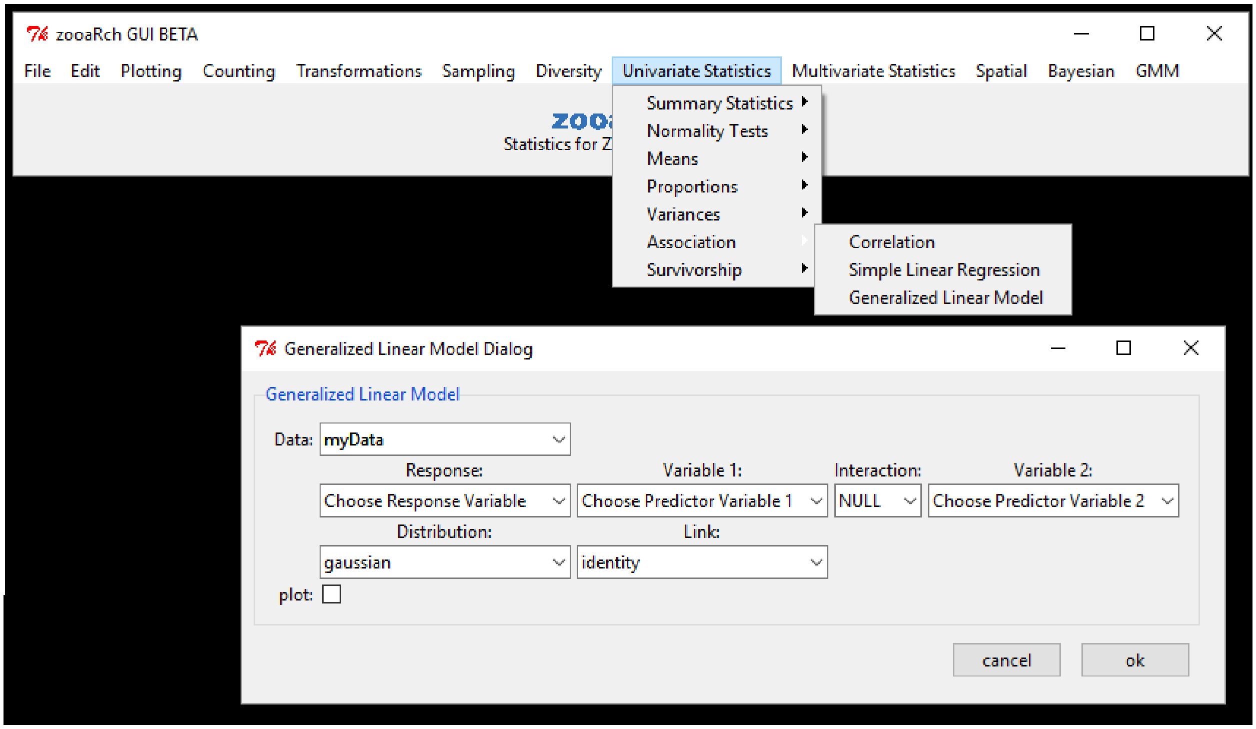
Response: Choose Response Variable

Predictor Variable: Choose Predictor Variable

plot: □

This function will perform a simple linear regression on the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The response variable and the predictor variable are chosen using the "Response" and "Predictor" comboboxes respectively. The results can be plotted if the "plot" checkbox is marked. The results are printed in the console.

Generalized Linear Model



Data: myData

Response: Choose Response Variable

Variable 1: Choose Predictor Variable 1

Interaction: NULL

Variable 2: Choose Predictor Variable 2

Distribution: gaussian

Link: identity

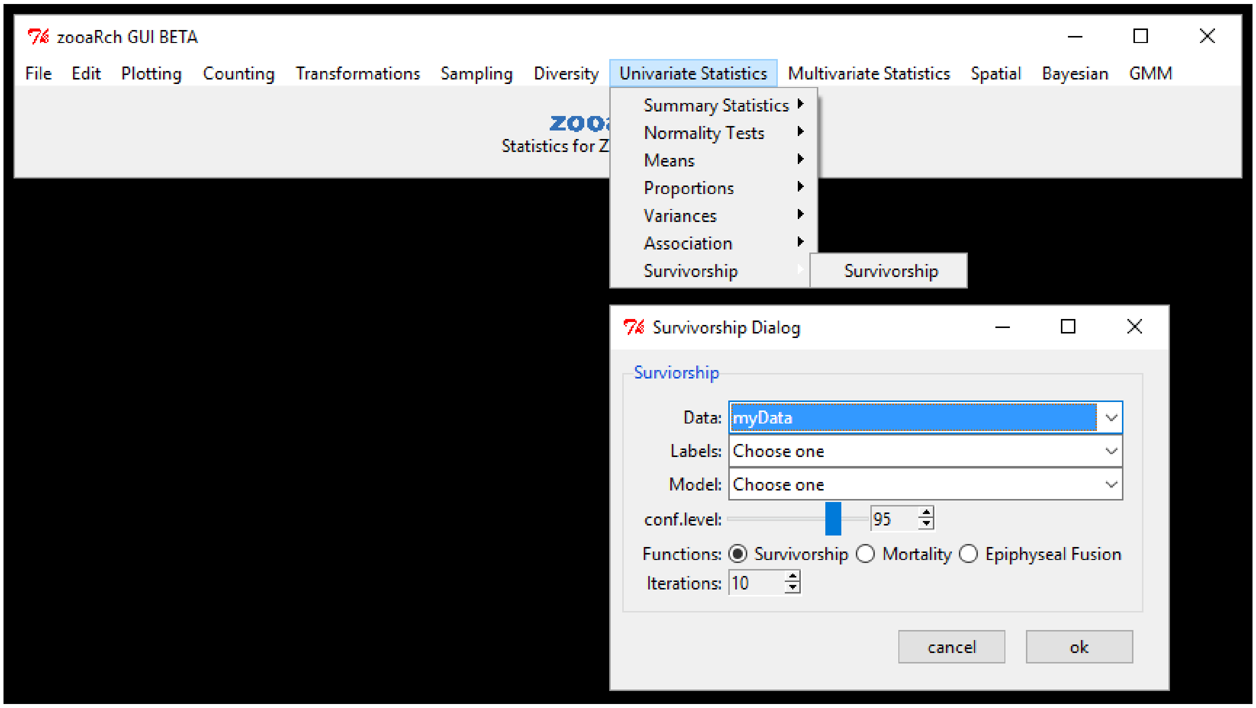
plot: □

This function will create a generalized linear model of the selected dataset. It will accept dataframes either already in R memory or loaded by the user. The response variable is selected using the "Response" combobox.

The first and second variable are selected using the “Variable 1” and “Variable 2” comboboxes respectively. The interaction between the variables can either be multiplication or addition depending on the selection in the interaction chosen in the “interaction” combobox. The distribution is determined by the “Distribution” combobox. The distribution can be either gaussian, binomial, poisson, or gamma. The variables can be linked either by identity, logit, log, or inverse. The results can be plotted if the “plot” checkbox is marked. The results are printed in the console and saved in R memory as “Results”.

8.7 Survivorship

Survivorship



Data: myData

Labels: Choose one

Model: Choose one

conf.level: 95

Functions: ● Survivorship ○ Mortality ○ Epiphyseal Fusion

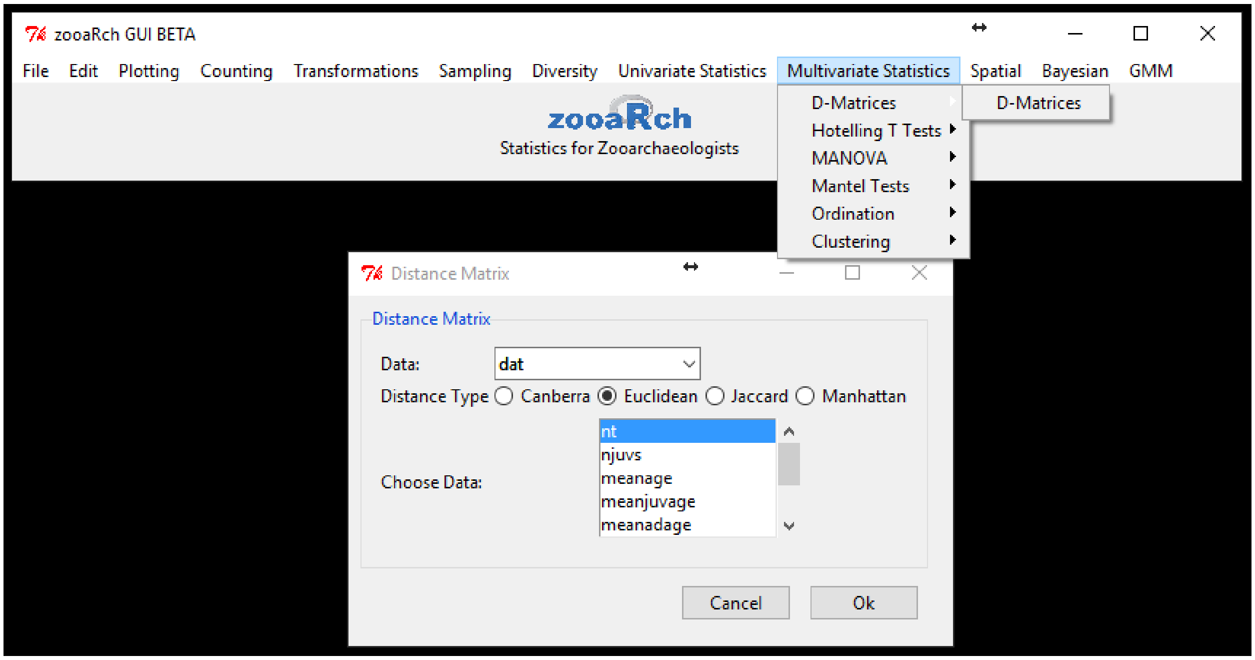
Iterations: 10

This function is will determine the survivorship of the dataset. It will accept dataframes either already in R memory or loaded by the user. The labels can be chosen using the “Labels” combobox to determine the dataset that the labels come from. The dataset that the model will come from is chosen using the “Model” combobox. The conf.level is selected using the conf.level slider. The functions that can be used are either the survivorship, Mortality, or Epophyseal Fusion functions. The number of iterations that the functions are run is determined by the “Iterations” spinbox. The results are printed in the console and saved in R memory as “Results”.

9. Multivariate Statistics Menu

9.1 D-Matrices

D-Matrices



Distance Matrix

Data: dat

Distance Type ○ Canberra ● Euclidean ○ Jaccard ○ Manhattan

Choose Data: nt, njuvs, meanage, meanjuvage, meanadage

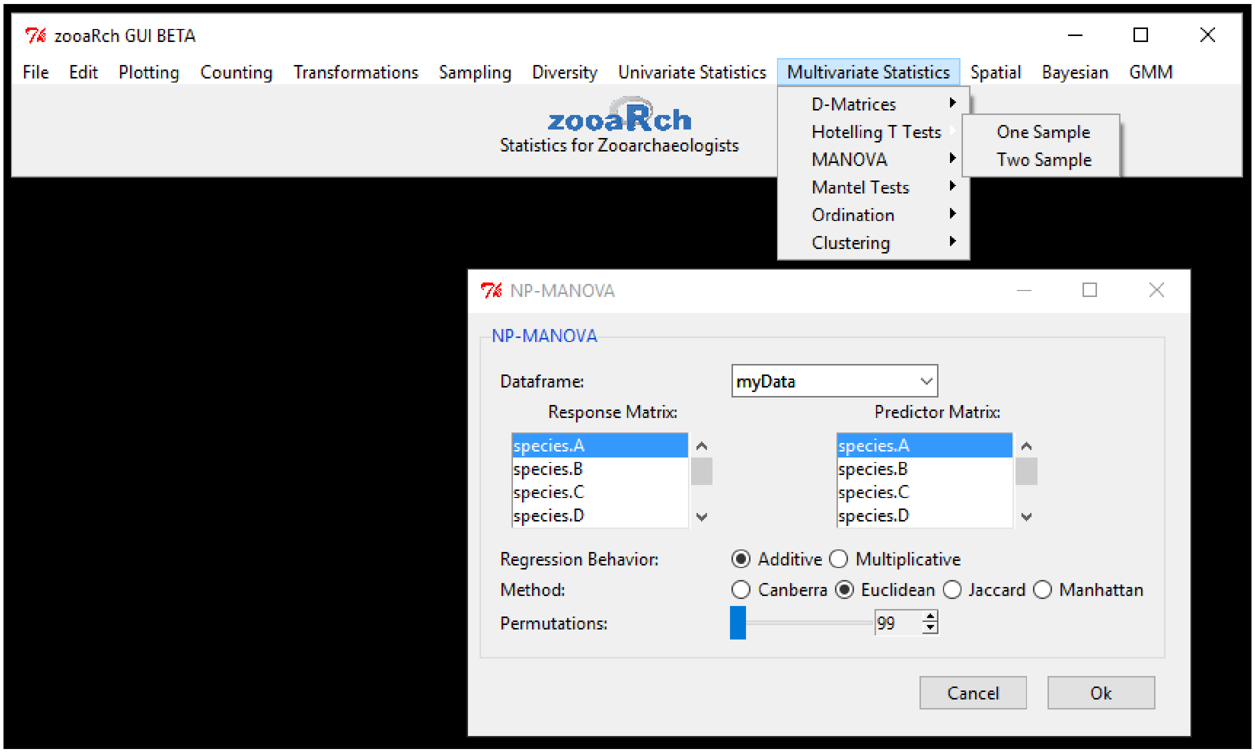
This function will calculate the distance function of the selected data. It will accept dataframes already in R memory or loaded by the user. The methods that the distance can be calculated are: Canberra, Euclidean, Jaccard, and Manhattan. The variables that the distance between will calculated are selected by the user. The output will be saved as a distance matrix as “D_Matrix_Results”.

9.2 Hotelling T-Tests

One Sample

This function is currently unavailable.

Two Sample*



NP-MANOVA

Dataframe: myData

Response Matrix: species.A, species.B, species.C, species.D

Predictor Matrix: species.A, species.B, species.C, species.D

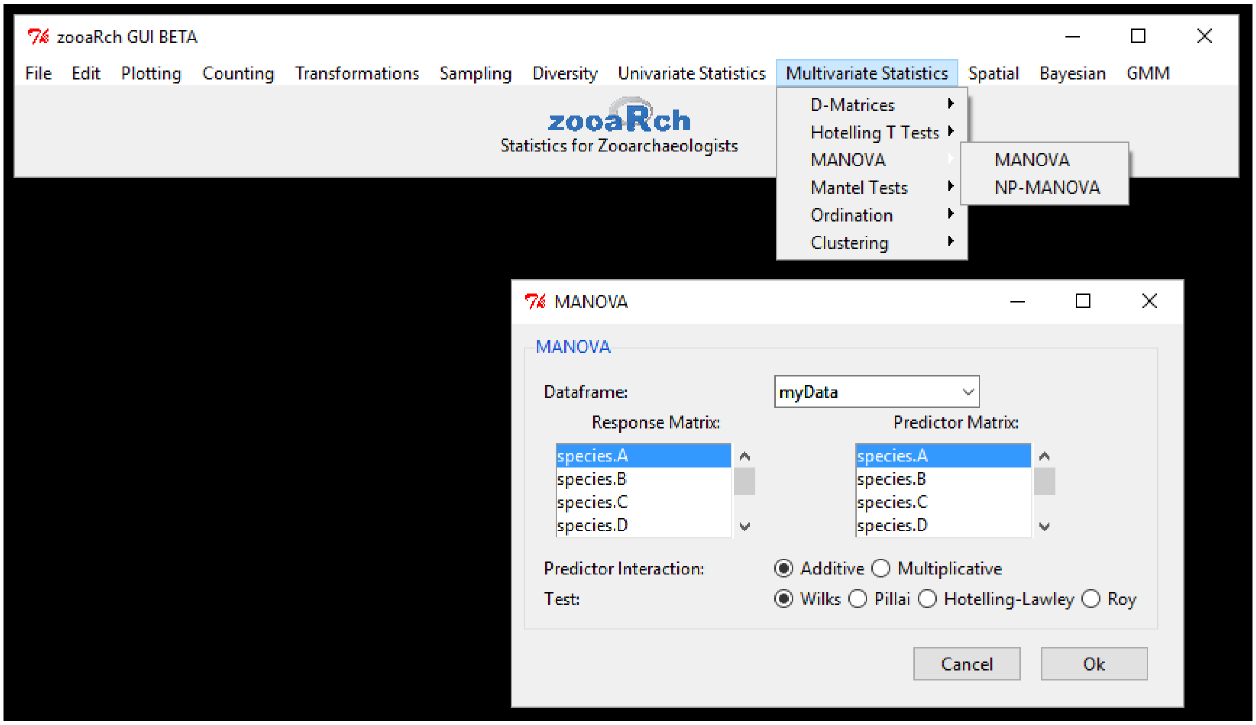
Regression Behavior: ● Additive ○ Multiplicative

Method: ○ Canberra ● Euclidean ○ Jaccard ○ Manhattan

Permutations: 99

9.3 MANOVA

MANOVA



MANOVA

Dataframe: myData

Response Matrix: species.A, species.B, species.C, species.D

Predictor Matrix: species.A, species.B, species.C, species.D

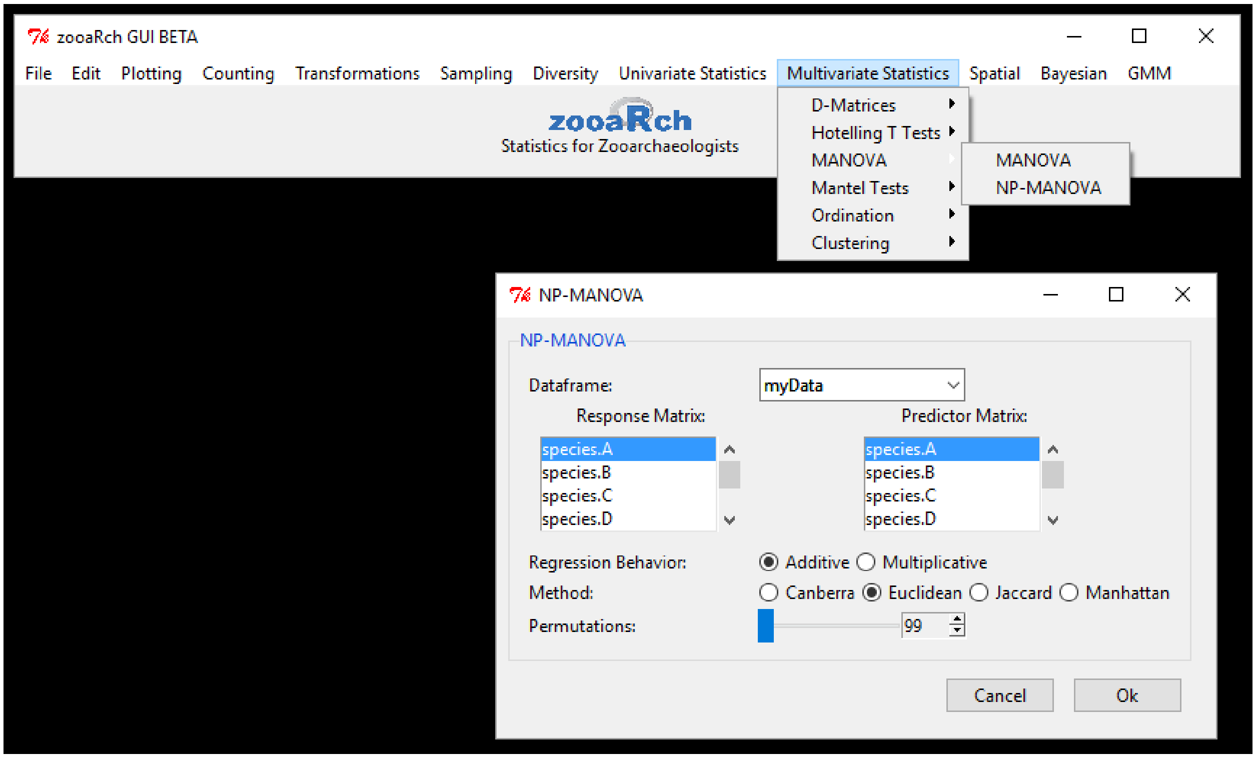
Predictor Interaction: ● Additive ○ Multiplicative

Test: ● Wilks ○ Pillai ○ Hotelling-Lawley ○ Roy

This function will perform a MANOVA on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The response and predictor variables are chosen using the “Response” and

"Predictor" comboboxes respectively. The interactions between the two variables can either be additive or multiplicative. The test that will be conducted can be the Wilks, Pillai, Hotelling-Lawley, Roy. The results are printed in the console and saved in R memory as "Results".

NP-MANOVA



Dataframe: myData

Response Matrix: species.A, species.B, species.C, species.D

Predictor Matrix: species.A, species.B, species.C, species.D

Regression Behavior: ● Additive ○ Multiplicative

Method: ○ Canberra ● Euclidean ○ Jaccard ○ Manhattan

Permutations: 99

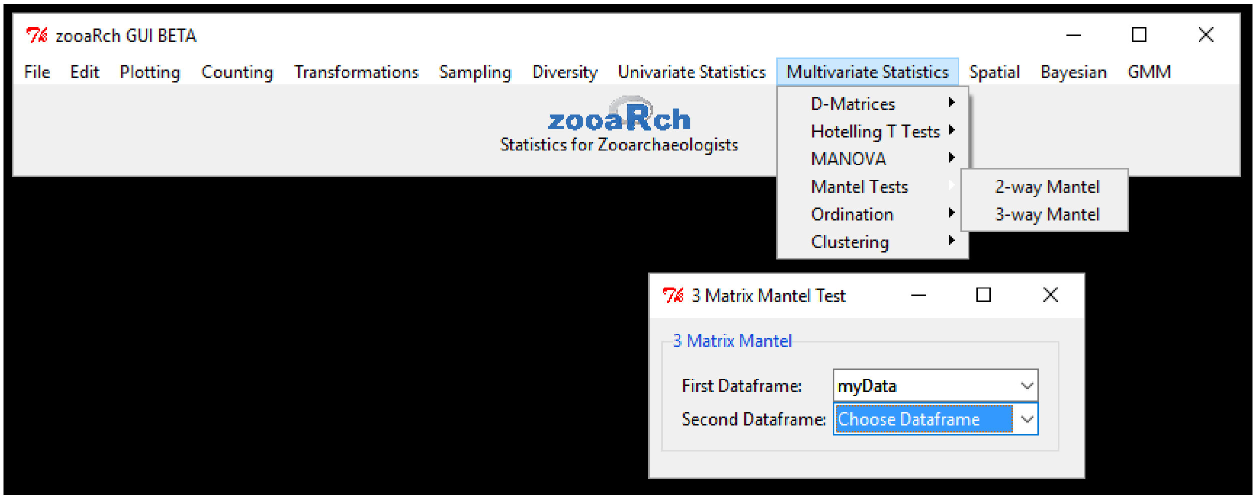
This function will perform a NP-MANOVA on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The response and predictor variables are chosen using the "Response" and "Predictor" comboboxes respectively. The interactions between the two variables can either be additive or multiplicative. The methods that will be used to find the distance can be the Canberra, Euclidean, Jaccard, or Manhattan. The number of permutations that is selected using the "Permutations" slider. The results are printed in the console and saved in R memory as "Results".

9.4 Mantel Tests

2-way Mantel*

This function will perform a 2-way Mantel test on the selected datasets. It will accept dataframes already in R memory or loaded by the user. The selected dataframes cannot contain missing values or characters. This function requires two dataframes that will be selected using their respective comboboxes. The columns that the test will be conducted on are selected using the listboxes that correspond to each of the dataframes. Multiple columns can be selected from each listbox. The distance method for each of the datasets can be either Canberra, Euclidean, Jaccard, or Manhattan. The number of permutations for the test is selected using the "Permutations" slider. The results will be printed in the console and saved in R memory as "Results".

3-way Mantel



First Dataframe: myData

Second Dataframe: Choose Dataframe

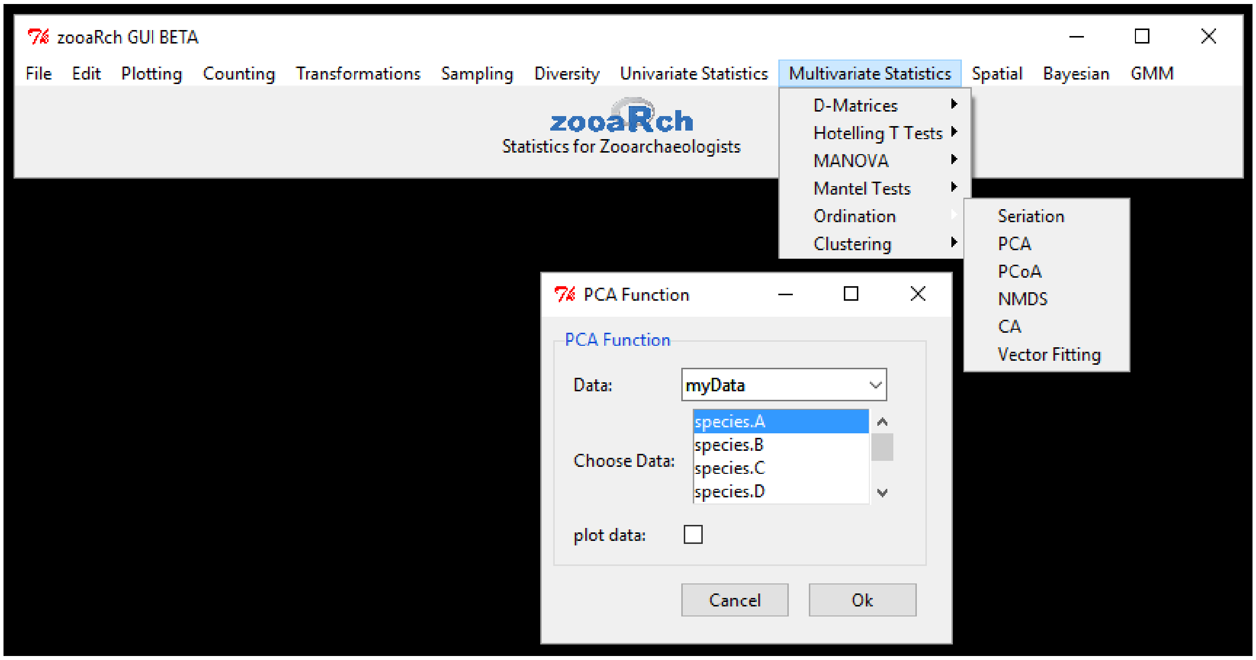
This function will perform a 3-way Mantel test on the selected datasets. It will accept dataframes already in R memory or loaded by the user. The selected dataframes cannot contain missing values or characters. This function requires three dataframes that will be selected using their respective comboboxes. The columns that the test will be conducted on are selected using the listboxes that correspond to each of the dataframes. Multiple columns can be selected from each listbox. The distance method for each of the datasets can be either Canberra, Euclidean, Jaccard, or Manhattan. The number of permutations for the test is selected using the "Permutations" slider. The results will be printed in the console and saved in R memory as "Results".

9.5 Ordinations

Seriation

This function is currently unavailable.

PCA



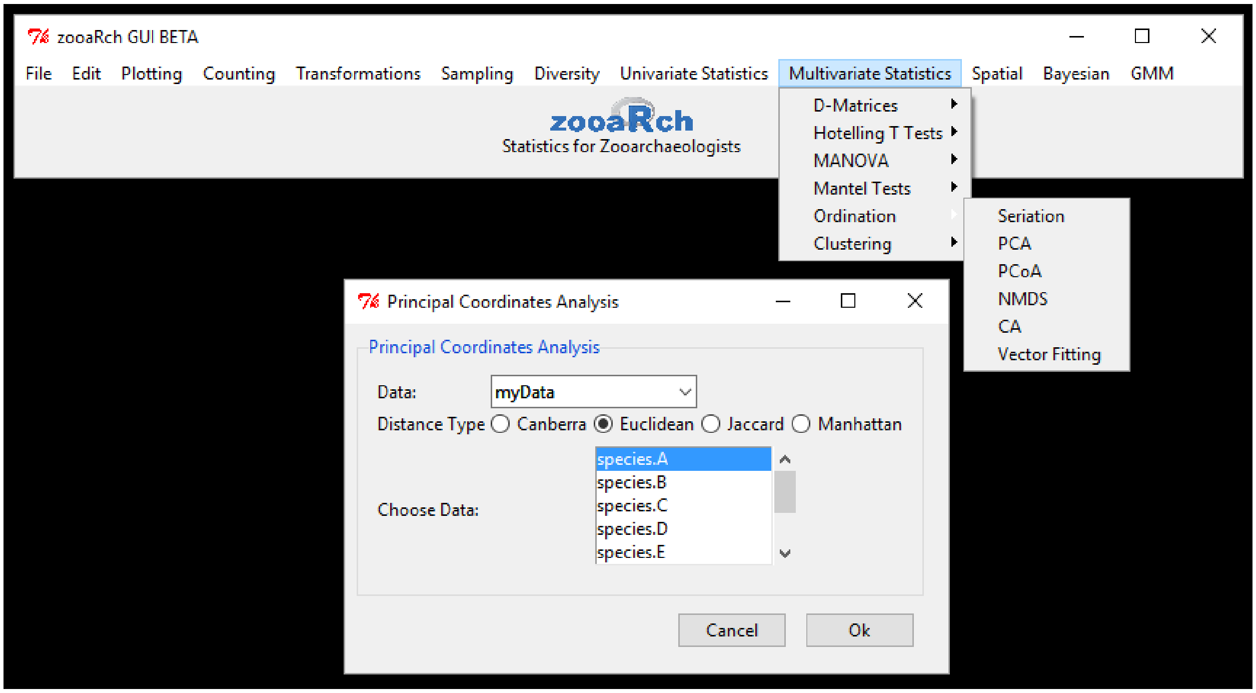
Data: myData

Choose Data: species.A, species.B, species.C, species.D

plot data: □

This function will perform a PCA test on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The selected dataframe cannot contain missing values or characters. The columns that the test will be performed on are selected using the “Choose Data” listbox. Multiple columns can be selected from the dataset. The results can be plotted if the “plot data” checkbox is marked. The results will be printed in the console and saved in R memory as “Results”.

PCoA



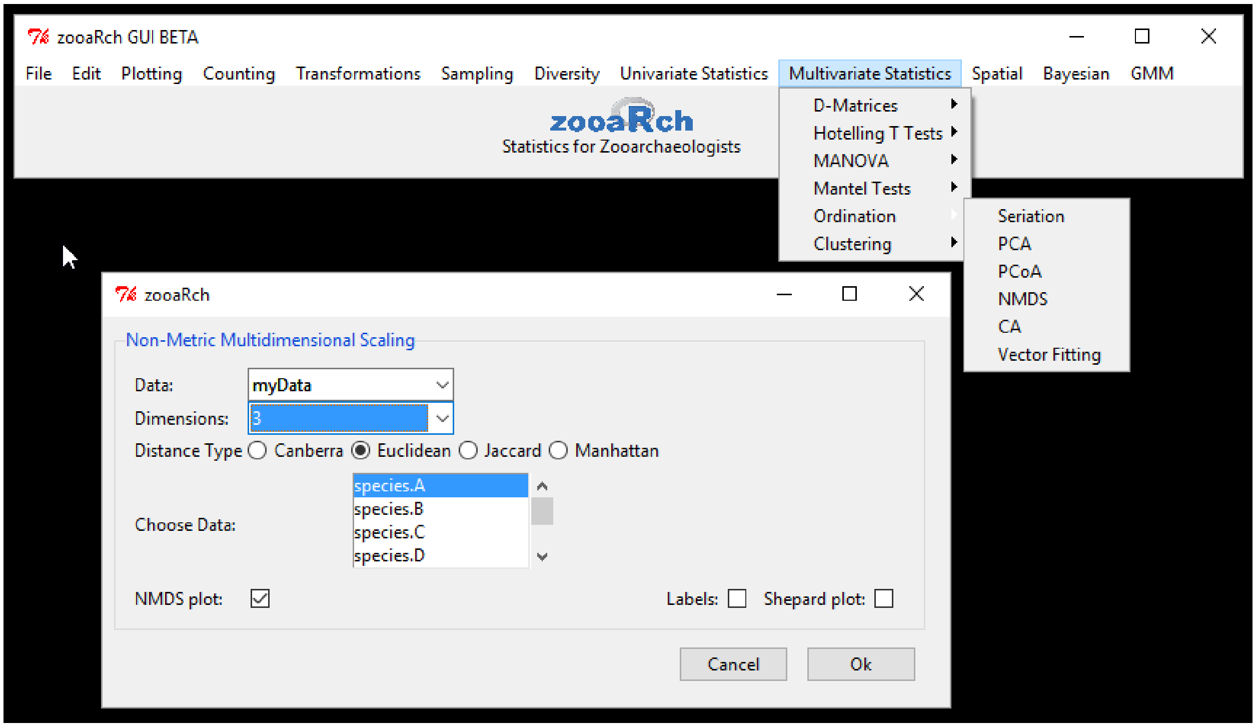
Data: myData

Distance Type ○ Canberra ● Euclidean ○ Jaccard ○ Manhattan

Choose Data: species.A species.B species.C species.D species.E

This function will perform a principle coordinate analysis on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The selected dataframe cannot contain missing values or characters. The distance method that will be used for the analysis can be either Canberra, Euclidean, Jaccard, or Manhattan. The columns that will be used are selected using the “Choose Data” listbox. Multiple columns can be selected from the dataset. The results of the analysis are plotted and printed in the console. They are also saved in R memory as “PCoA_Results”.

NMDS



Non-Metric Multidimensional Scaling

Data: myData

Dimensions: 3

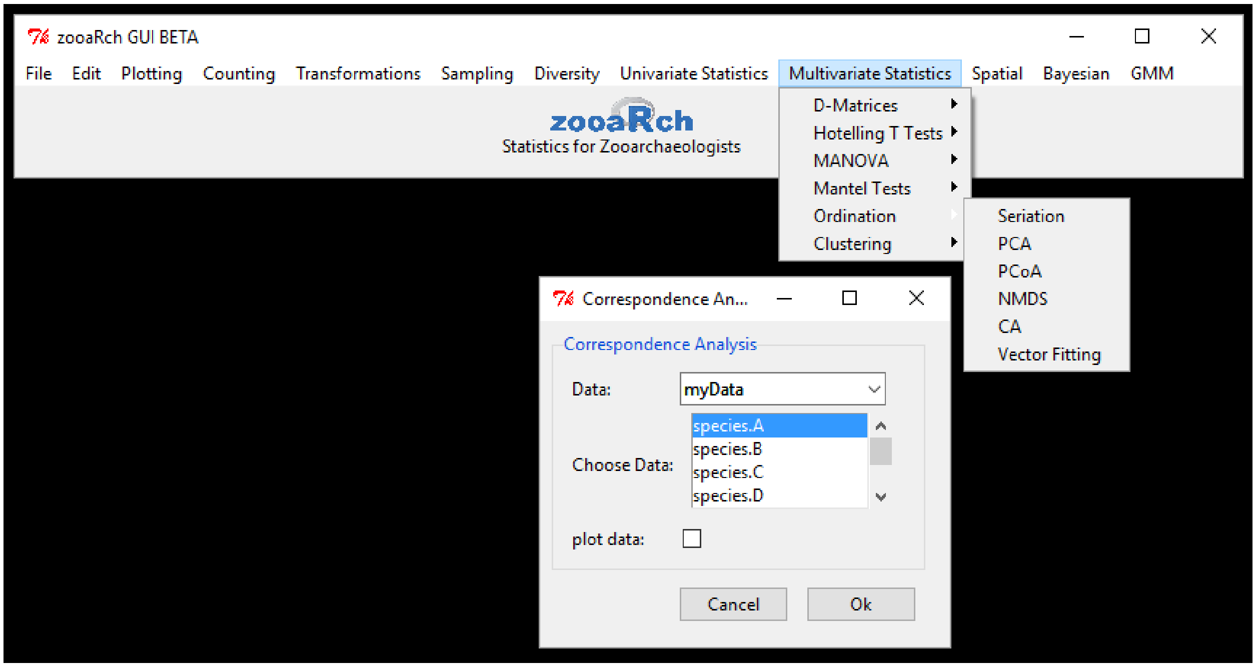
Distance Type ○ Canberra ● Euclidean ○ Jaccard ○ Manhattan

Choose Data: species.A, species.B, species.C, species.D

NMDS plot: ☑ Labels: □ Shepard plot: □

This function will perform a non-metric multidimensional scaling analysis on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The selected dataset cannot contain missing values or characters. The number of dimensions that the analysis will be conducted in is selected using the “Dimensions” combobox. The method that the distance will be calculated in can be either Canberra, Euclidean, Jaccard, or Manhattan. The columns that the analysis will be conducted on are chosen using the “Choose Data” listbox. Multiple columns can be chosen from the dataset. A NMDS plot will be created if the “NMDS plot” checkbox is marked. The plot will contain labels if the “Labels” checkbox is marked. A Shepard plot will be created if the “Shepard plot” checkbox is marked.

CA



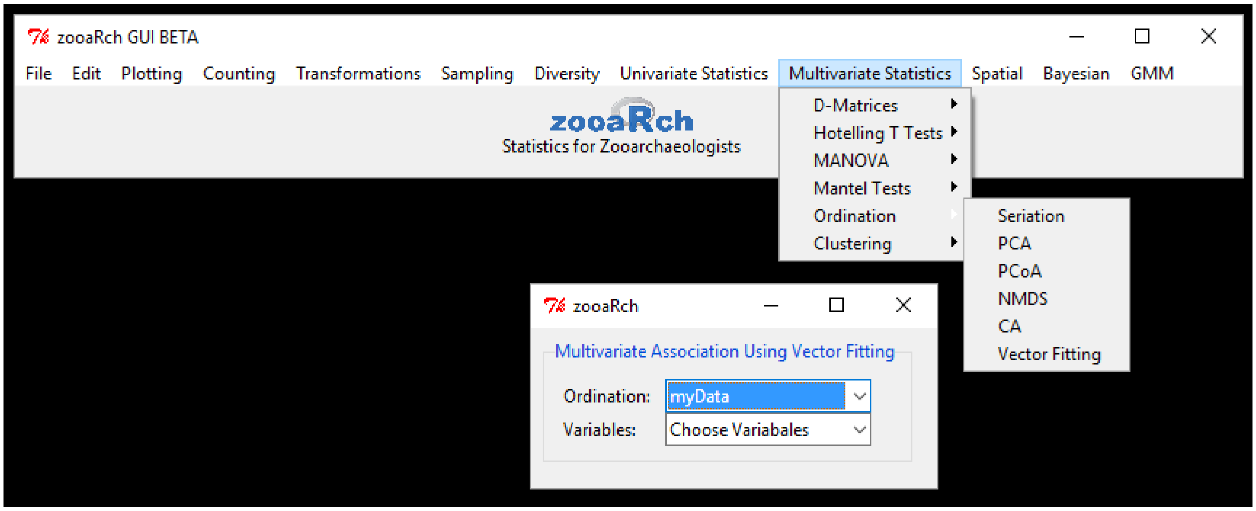
Data: myData

Choose Data: species.A, species.B, species.C, species.D

plot data: □

This function will perform a correspondence analysis on the selected dataset. It will accept dataframes already in R memory or loaded by the user. The selected dataset must not contain missing values or characters. The columns that will be used in the analysis are selected using the “Choose Data” listbox. The results can be plotted if the “plot data” checkbox is marked. The results of the analysis are printed in the console and saved in R memory as “Results”.

Vector Fitting



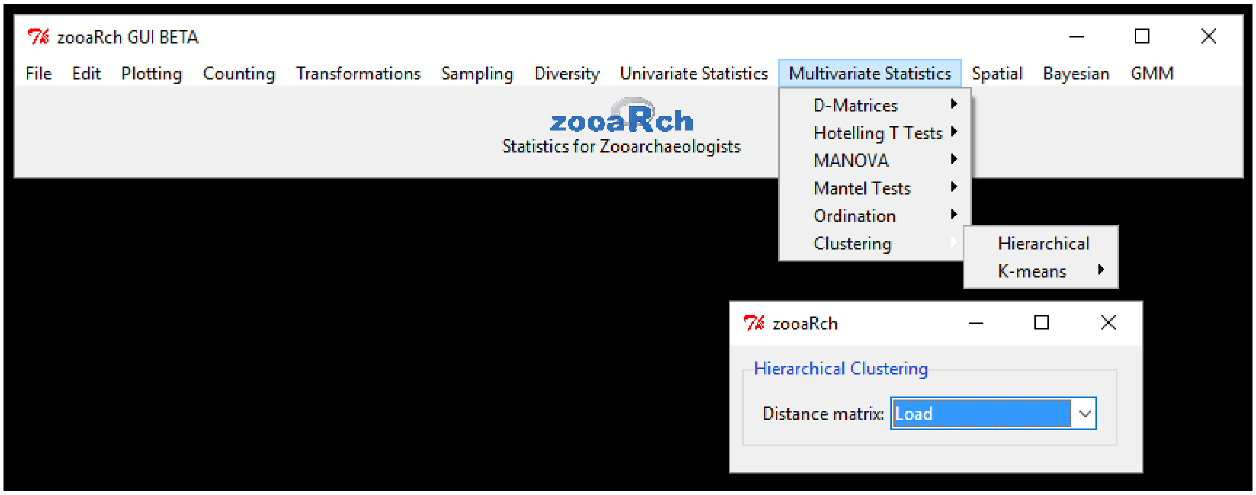
Ordination: myData

Variables: Choose Variabales

This function will perform a multivariate association analysis using *vector fitting*. It will accept matrices already in the R memory or loaded by the user. The selected dataset must *be the results of an ordination (PCA, CA, PCoA, NMDS)*. The ordination matrix is chosen using the “Ordination” listbox. The variables to be regressed against the ordination should be contained within a matrix object. The matrix of variables will be selected using the “Variables” listbox. The number of axes that the analysis will use with is chosen using the “N Axes” listbox. The number of permutations is chosen using the “Permutations” slider. The results of the analysis will be plotted and printed in the console. They will be saved in R memory as “Vectorfit_Results”.

9.6 Clustering

Hierarchical



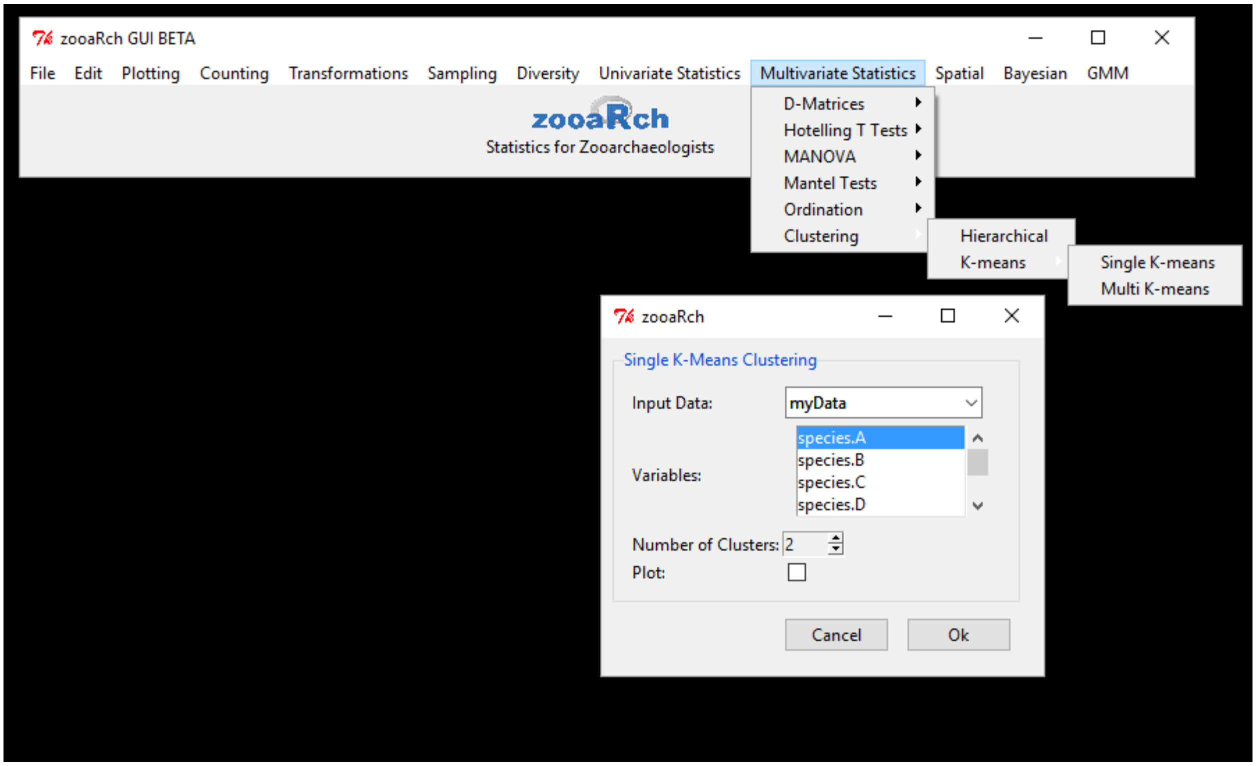
Hierarchical Clustering

Distance matrix: Load

This function will perform various hierarchical clustering methods on the selected dataset. It will accept distance matrices from the R memory or loaded by the user. The clustering method can be either Single linkage, Complete linkage, Average linkage, Centroid linkage, Mcquitty, Median linkage, Ward's method, or Ward's 2. The results are plotted.

K-means

Single K-means



Single K-Means Clustering

Input Data: myData

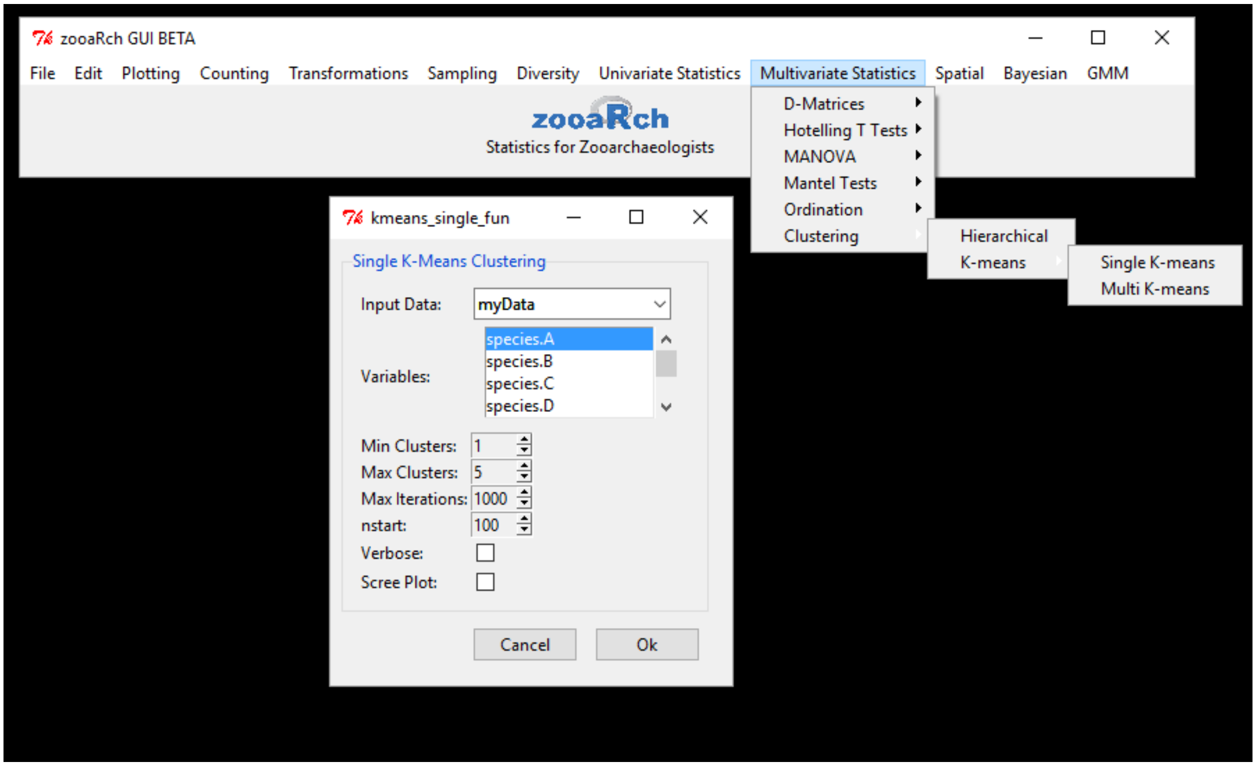
Variables: species.A, species.B, species.C, species.D

Number of Clusters: 2

Plot: □

This function will perform K-means clustering using simplified parameters. It will accept distance matrices from the R memory or loaded by the user. The selected dataset cannot contain missing values or characters. The columns that will be used are selected using the “Variables” listbox. Multiple columns can be chosen from the dataset. The number of clusters can be chosen using the “Number of Clusters” spinbox. The results can be plotted if the “Plot” checkbox is marked. The results saved in R memory as “SingleKmeans”.

Multiple K-means



Input Data: myData

Variables: species.A, species.B, species.C, species.D

Min Clusters: 1

Max Clusters: 5

Max Iterations: 1000

nstart: 100

Verbose: □

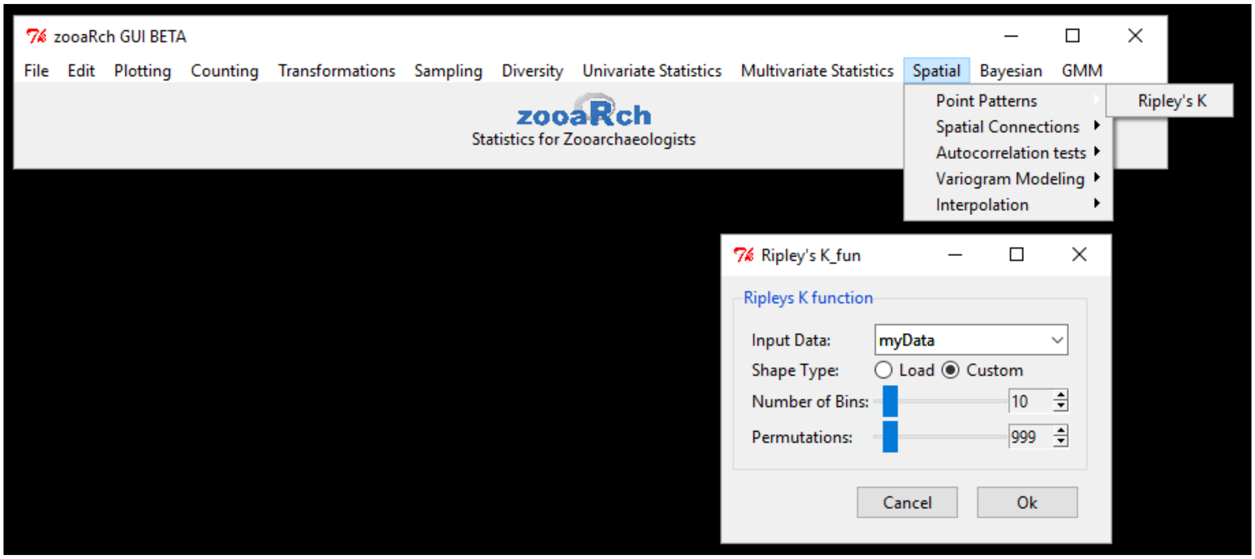
Scree Plot: □

This function will perform K-means clustering using user-inputted parameters. It will accept distance matrices from the R memory or loaded by the user. The dataset cannot contain missing values or characters. The columns that will be used are selected using the “Variables” listbox. Multiple columns can be chosen from the dataset. The minimum and maximum number of clusters can be chosen by the “Min Clusters” and “Max Clusters” spinboxes respectively. The Max number of iterations and the number of random sets that can be chosen are selected by the “Max Iterations” and “nstart” spinboxes respectively. The function will be run verbose if the “Verbose” checkbox is marked. A Scree plot will be created if the “Scree Plot” checkbox is marked. The results will be saved in R memory as “MultiKmeans”.

10. Spatial Statistics Menu

10.1 Point Patterns

Ripley’s K



Input Data: myData

Shape Type: ○ Load ● Custom

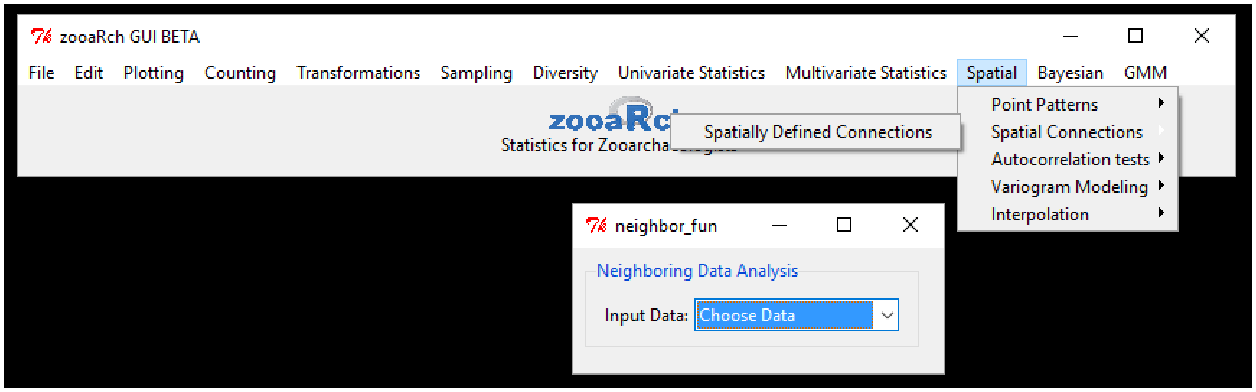
Number of Bins: 10

Permutations: 999

This function will perform a Ripley's K analysis on the selected dataset. It will accept dataframes already in R memory or loaded by the user. It will also accept shapefiles loaded by the user. The shape of the boundary box of the Ripley's K function can be loaded as a shapefile by selecting the "Load" option. The loaded shapefile cannot contain more than one polygon. The boundary shape can also be a custom draw figure if the "Custom" option is selected. With the custom shape a figure will appear on the console. The user will select the vertices of the shape that they wish to create. When the shape is complete the user will click finish to create the shape. The number of bins that the Ripley's K function will use is selected using the "Bins" slider. The number of permutations used is determined by the "Permutations" slider. The results of the function will be plotted and printed in the console.

10.2 Spatial Connections

Spatially Defined Connections

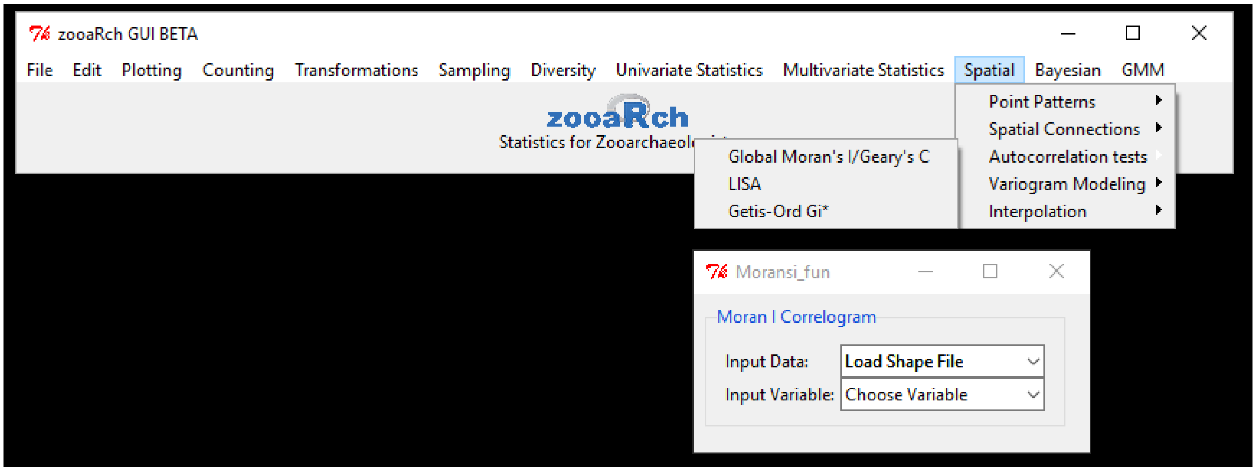


Input Data: Choose Data

This function will find spatially defined connections in the selected dataset. It will accept spatial polygon dataframes from the R memory or loaded by the user. It will also accept shapefiles loaded by the user. The spatially defined connections can be found using the Queen, Rook, Delauney, Gabriel, or Relative methods. The voroni data can be plotted if the "Voroni data" checkbox is marked. The results of the function are plotted.

10.3 Autocorrelation Tests

Global Moran's I/Geary C



Input Data: Load Shape File

Input Variable: Choose Variable

This function will perform a Global Moran's I or Geary C test. It will accept spatial polygon dataframes from the R memory or loaded by the user. It will also accept shapefiles loaded by the user. The input variable is a list in the loaded file that the test will be conducted with. The display title will be the title of the created plot. The method that the test will use is either Moran's I or Geary C. The number of bins is determined by the "Number of Bins" slider. The results of the test are plotted.

LISA

This function is currently unavailable.

Getis ORD GI*

This function is currently unavailable.

10.4 Variogram Modeling

Plot Spatial Covariance

This function is currently unavailable.

Plot Spatial Semi-Variogram

This function is currently unavailable.

Fit Semi-Variogram

This function is currently unavailable.

10.5 Interpolation

Kriging

This function is currently unavailable.

TPS

This function is currently unavailable.

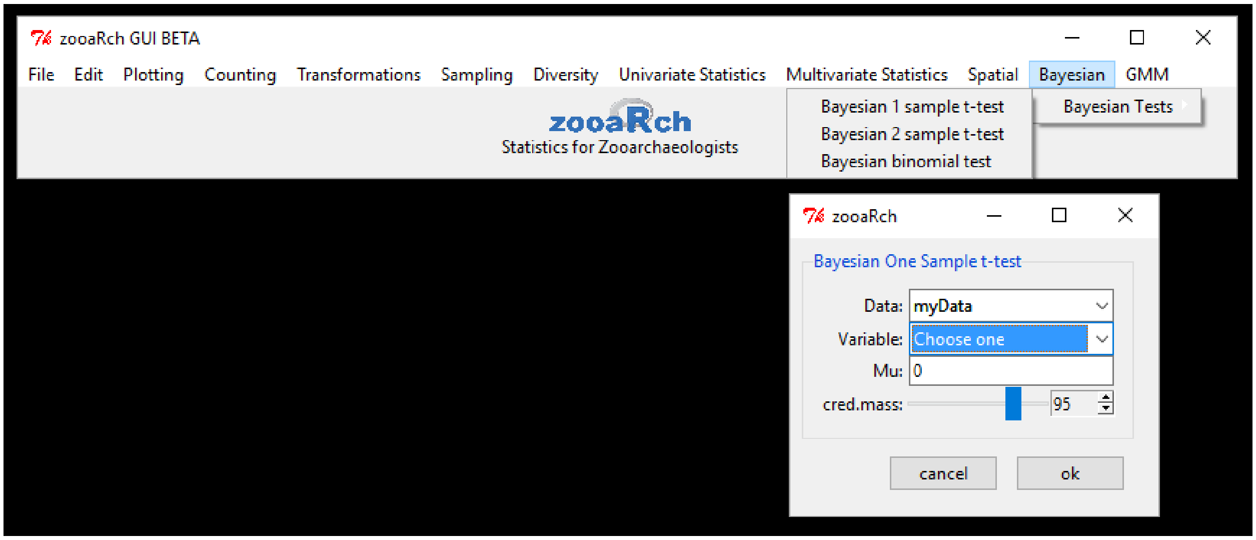
IDW

This function is currently unavailable.

11. Bayesian Statistics Menu

11.1 Bayesian Tests

Bayesian 1 sample t-test



Data: myData

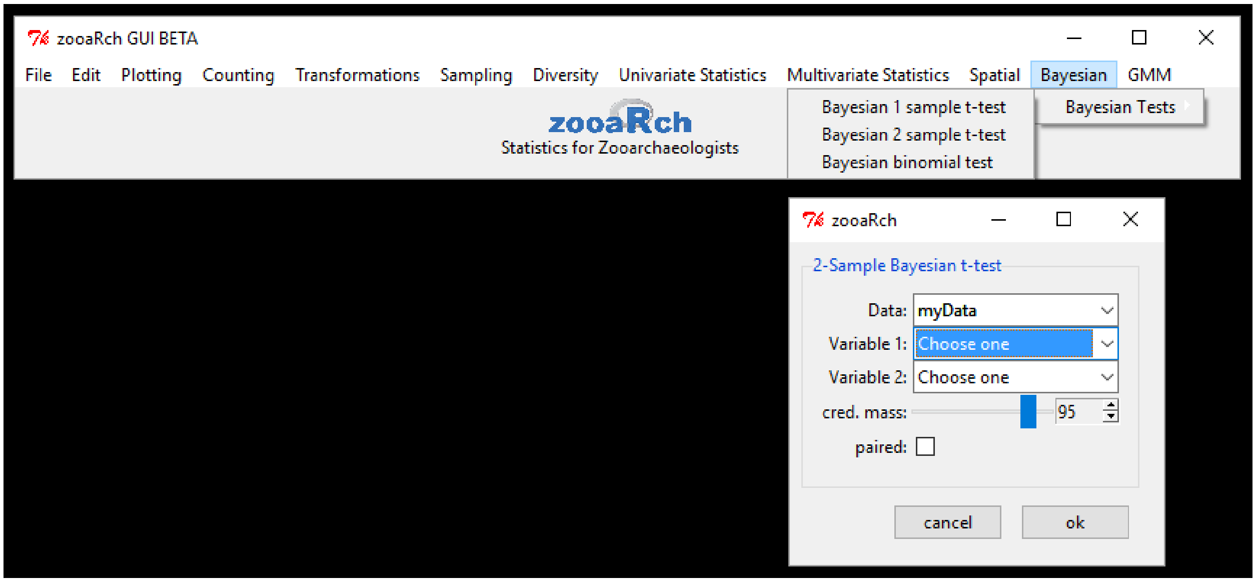
Variable: Choose one

Mu: 0

cred.mass: 95

This function will perform a Bayesian one sample t-test using the selected data. It will accept dataframes already in R memory or loaded by the user. The column that the test will be performed with is selected using the “Variable” combobox. The Mu value the number in the “Mu” entrybox. The credible interval level is determined by the “cred.mass” slider. The results are printed in the console and saved in R memory as “Results”.

Bayesian 2 sample t-test



Data: myData

Variable 1: Choose one

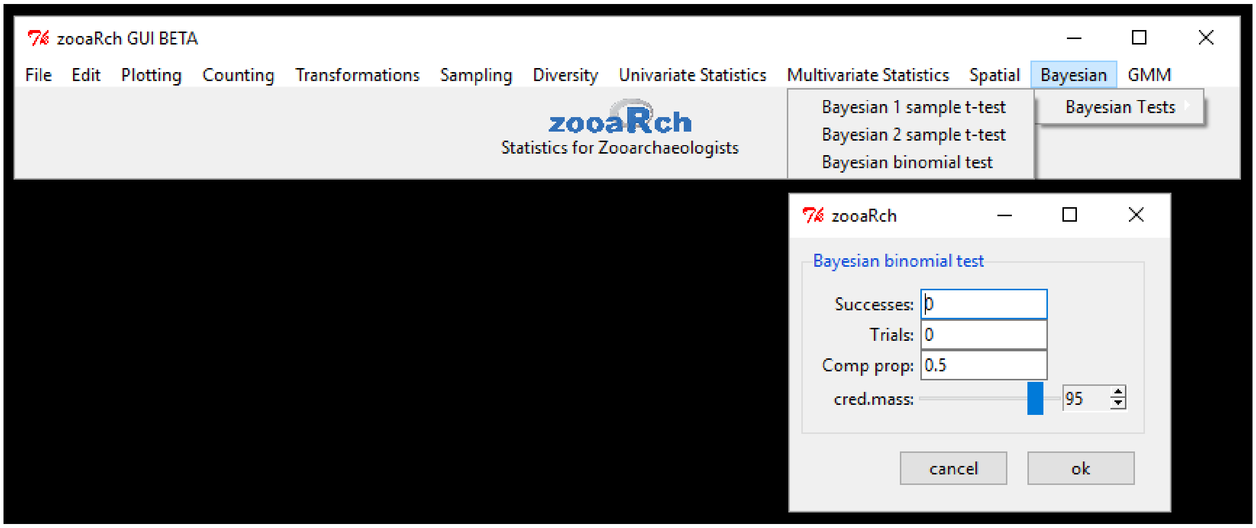
Variable 2: Choose one

cred. mass: 95

paired: □

This function will perform a Bayesian two sample t-test using the selected data. It will accept dataframes already in R memory or loaded by the user. The two columns that will be performed with is selected using the "Variable 1" and "Variable 2" comboboxes respectively. The credible interval is determined by the "cred.mass" slider. The test will be performed as a paired test if the "paired" checkbox is marked. The results are printed in the console and saved in R memory as "Results".

Bayesian Binomial Test



Successes: 0

Trials: 0

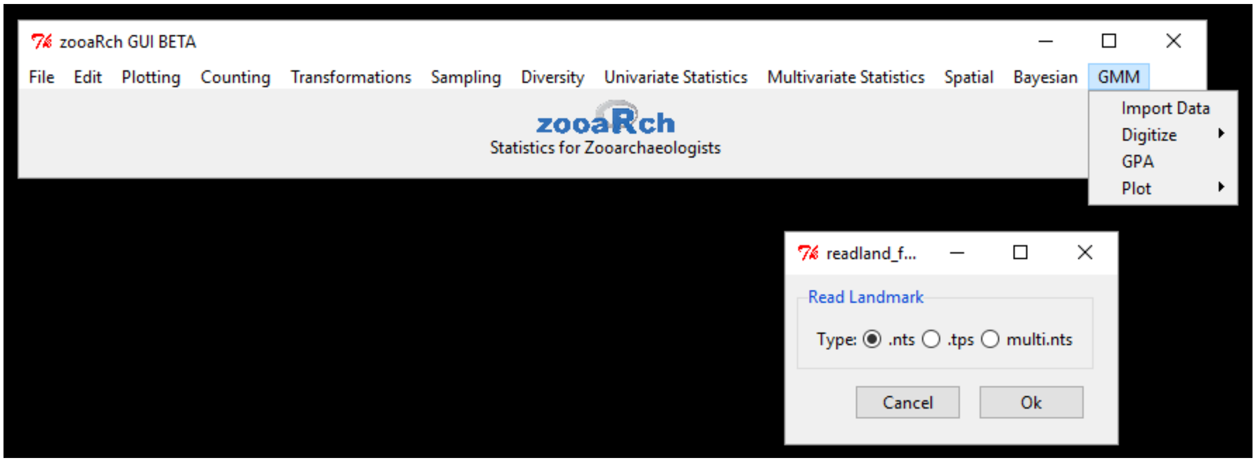
Comp prop: 0.5

cred.mass: 95

This function will perform a Bayesian Binomial test on the selected data. The number of successes and trials that will be performed based on the values in the "Successes" and "Trials" entryboxes. The composition proportion is the value in the "Comp prop" entrybox. The credible interval is determined by the "cred.mass" slider. The results of the test are printed in the console and saved in R memory as "Results".

12. Geometric Morphometrics Menu

12.1 Import Data


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
Import Data
Digitize
GPA
Plot
zooaRch
Statistics for Zooarchaeologists
readland_f...
Read Landmark
Cancel
Ok

Type: ● .nts ○ .tps ○ multi.nts

This function will import .tps .nts, or multiple .nts files. The imported data will be saved as an array with the name of the original file in R memory unless the multi.nts option is chosen. Then the data will be saved in R memory as “collective.array”.

12.2 Digitize

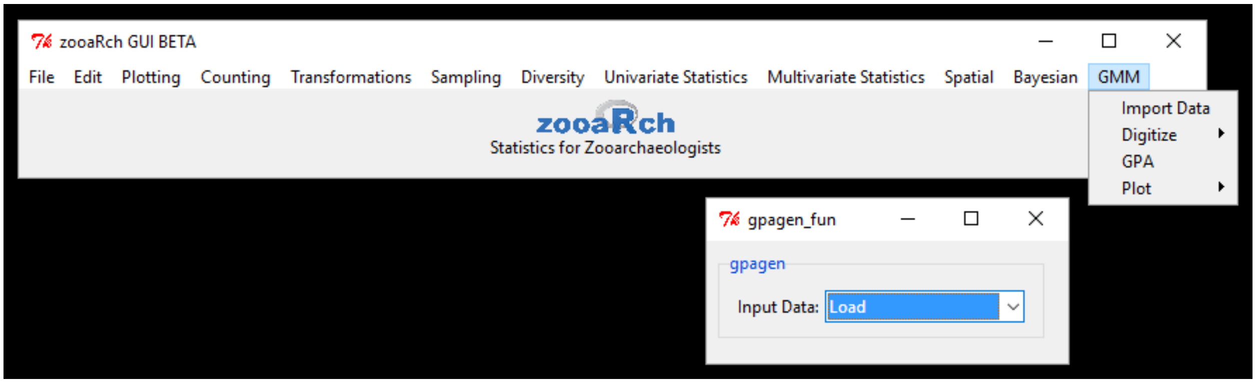
2D-Digitize

This function is currently unavailable.

3D-Digitize

This function is currently unavailable.

12.3 GPA


zooaRch GUI BETA
File Edit Plotting Counting Transformations Sampling Diversity Univariate Statistics Multivariate Statistics Spatial Bayesian GMM
Import Data
Digitize
GPA
Plot
zooaRch
Statistics for Zooarchaeologists
gpagen_fun
gpagen

Input Data: Load

This function will perform a gpagen analysis on the selected dataset. It will accept arrays from the R memory or loaded by the user. The curve and the surface arrays are selected from the R memory or loaded by the user. If “None” is selected, then there will be nothing inputted for those values. The type of test that is used is either the Prod, or Bending Energy. The progress is plotted if the “Plot Progress” checkbox is marked. The results of the test are printed in the console, plotted and saved in R memory as “gparesults”.

12.4 Plot

Tangent Space

This function is currently unavailable.

Deformation

This function is currently unavailable.

Reference-Target Deformation

This function is currently unavailable.

Analysis

This function is currently unavailable.