Bat Translation Script Guide

Contents

[What You’ll Need: 3](#_Toc185227622)

[Your data.csv 3](#_Toc185227623)

[Species\_Translations.csv 4](#_Toc185227624)

[Guilds.csv 5](#_Toc185227625)

[Setting up Directories 6](#_Toc185227626)

[Sourcing “Main.R” 7](#_Toc185227627)

[Data Folder 7](#_Toc185227628)

[Operating the program: 8](#_Toc185227629)

[Final Output: 10](#_Toc185227630)

# What You’ll Need:

* Your\_data.csv (Or whatever it happens to be called)
* Species\_Translations.csv
* Guilds.csv

### Your data.csv

The data received from the bat recorder. Which should be formatted something like this:

A screenshot of a table

Description automatically generated

The way the script is written requires there to be at least some variables and that they are named as such, but feel free to alter this to your needs:

A computer screen with text

Description automatically generated

The “rework\_Time” function will combine the TIME and HOUR columns while removing any milliseconds or instances where the TIME column contains the hours. So if you have your data formatted differently, comment out this line within “Main.R”

Important!

The only bit of pre-processing that needs to be done is to make sure that in cases where multiple species exist in the same row, they are separated by either a comma or “and” like so:

A screenshot of a computer

Description automatically generated

If you forget, don’t worry. The program will let you ignore those translations and return to them later. However, rerunning the program on the outputted/formatted data is a bit annoying at the moment, so I’d recommend not skipping this step.

### Species\_Translations.csv

The Species\_Translations.csv file determines what choices the program gives the user for renaming all the manual IDs. It should be a single column of all translation names like so:

A screenshot of a computer

Description automatically generated

### Guilds.csv

The guilds.csv file is how the program determines what guilds and complexes each species should be placed into. It’s formatted in two columns, the species/complex name and its guild/group.

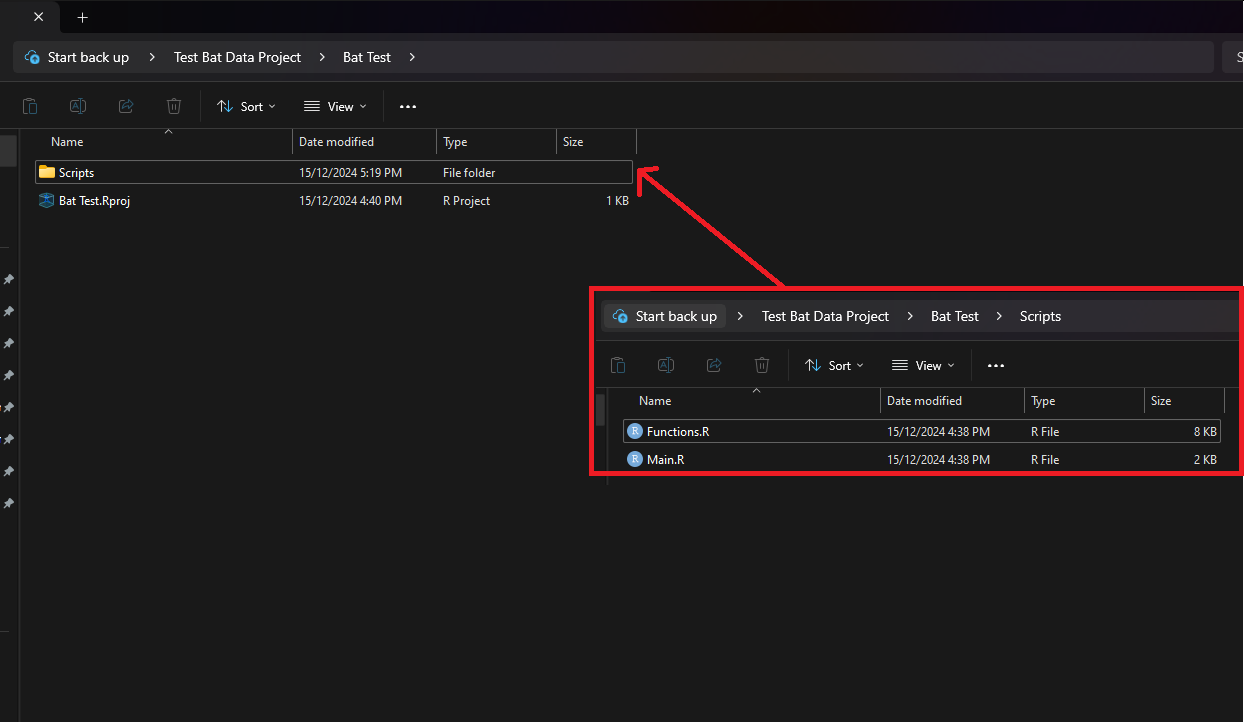
A screenshot of a computer

Description automatically generated

# Setting up Directories

The program consists of two scripts, “Main.R” and “Functions.R”.

Main.R will attempt to source Function.R from a scripts folder inside your working directory, so either manually adjust where it’s sourcing the file from or set up your folders like this:



You can also add a “Data” and “Outputs” folder if you’d like but the program will do that for you.

# Sourcing “Main.R”

When you source “Main.R” it will start by doing a few different things.  
If you haven’t made a “Data” or “Outputs” folder, it will do that for you and then prompt you to add your CSV to it.

There is some test data as well as the expected outputted data, in the compressed zip if you want to practice using the program before using it on a large data set.

### Data Folder

Your data folder should contain:

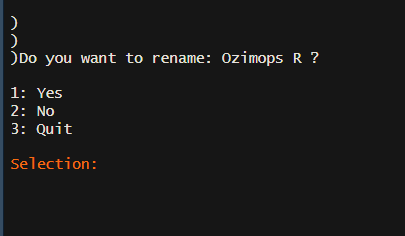
* Yourdata.csv
* Guilds.csv
* Species\_Translations.csv

A screenshot of a computer

Description automatically generated

### Operating the program:

The program runs by finding any Manual ID that isn’t within your species translations list and then changing it based on the choices within the species translations. When it finds an item not in the list it prompts the user if they’d like to change it.



Select your option by typing the number associated with your choice and pressing ‘Enter’

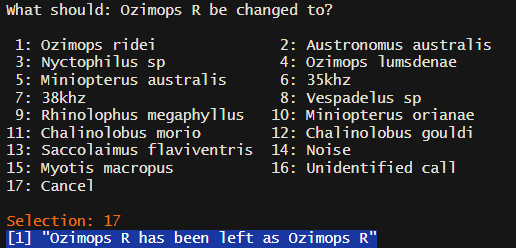
Yes – Will give you the option to change all items with that name to a name within your species translations list.

A screen shot of a computer

Description automatically generated

Again, just type the number associated with your choice and press ‘Enter’

No (or selecting cancel after selecting yes) – Will leave the item as the same name it was previously.



Quit – Stops translating items and prints the final CSV to the outputs folder.

Important !

Before the program outputs the final document, named “Bat\_Accoustic\_Recorder\_Data”, it will prompt you to close any open CSV files. This is important in the case where the program is trying to overwrite the original file. If this file is open, the program will fail to output, and nothing will be produced.

# Final Output:

Your final output will be written as a CSV in the Outputs folder within your working directory.

