Workshop Overview

An introduction to the ‘glatos’ R package

February 27, 2020

Organizers and instructors:

Chris Holbrook, Nancy Nate, Tom Binder, Darryl Hondorp, Jon Pye, Caitlin Bate, Ryan Gosse

Workshop materials: To be provided at the workshop.

Overview

The workshop will introduce tools available in the ‘glatos’ R package for loading, manipulating, summarizing, and visualizing acoustic telemetry data. The workshop will include code demonstrations by workshop leaders and exercises where attendees will apply and extend concepts to sample datasets or their own GLATOS data.

Perquisites

Attendees should have at least beginner-level knowledge of the R programming language. For example, attendees should (1) have completed an introductory R course or (2) be familiar with the concepts described in pages 1-59 of the document “An introduction to R for analyzing acoustic telemetry data” (PDF available here).

Preparation

Attendees should:

1. Bring a laptop with the following software installed (see installation instructions below). WARNING: Software installation may require administrative rights to your machine and may take considerable time, depending on internet connection. Install in the order shown so that RStudio will automatically find and link with R. If you have different versions of the software already installed on your computer, we recommend removing previously installed software and re-installing the recommended version to minimize the possibility of installation errors.

    a. R (V. 3.6.2)

    b. RStudio (V. 1.2.5033): A free and open source integrated development environment for R that simplifies code writing and debugging.

    c. GDAL (V. 3.0): An open source geospatial library for manipulating spatial data. It is an external program (not an R package) and is required for non-linear interpolation of fish tracks in the ‘glatos’ package.

    d. RTools (V. 3.5): A collection of resources for building R packages. It is an external program (not an R package) and is required to install the glatos R package from source code on gitlab.

    e. glatos R package (V. ≥ 0.4.0 full-english-breakfast): From source code on OTN gitlab.

    f. VTrack R package (V. 2.0.0): From source code on GitHub.

Instructions for installing R:

1. Point your browser to https://cran.r-project.org/mirrors.html and select one of the mirrors for the USA.

2. Click on the "Download" link of precompiled binary for the distribution that matches your operating system.

- If you are installing to a Windows computer, click on "Download R for Windows" link. Under subdirectories, click on “base” and then “Download R 3.6.2 for Windows.
- If you are installing to a Mac, click "Download R for (Mac) OS X" and on the resulting webpage click the "R-3.6.2.pkg" link to download the installer to your computer.

4. Once R finishes downloading, click on the installer and follow the prompts. Make sure to "install as administrator" if you are on Windows. Otherwise, the default options are fine.

5. After install finishes, you should be able to open a R command line terminal by selecting the icon in Windows, or through Launchpad in MacOS.

Instructions for installing RStudio:

1. Point your browser to https://www.rstudio.com/products/rstudio/download/ , click the Download button under the RStudio Desktop Free version and you will be directed to download the appropriate RStudio installer for your operating system.

2. Click on downloaded installer and follow the prompts for installing RStudio. Unless you have specific installation requirements, the default options are fine.

3. Once RStudio has finished installing, click on the program icon. RStudio should open with an active R console.

Instructions for installing GDAL

First, check if you already have GDAL installed. On Windows, open Command Prompt. On Mac, open Terminal (Applications/Utilities/Terminal). Type the following and hit enter:

```
    gdalinfo --version
```

If a version number is displayed, e.g., `GDAL 3.0.4, released 2020/01/28`, then you are done (GDAL is installed). Otherwise, following instructions below for your operating system.

Windows operating system

The easiest method for installing GDAL on windows is to download and install the OSGeo4W software package. OSGeo4W is a binary distribution of a set of open source geospatial software for Windows operating system. In addition to GDAL libraries, OSGeo4W includes QGIS and GRASS software and many other useful packages. A complete install of OSGeo4W will consume a substantial portion of hard drive space. However, only GDAL is required for the GLATOS workshop and may be selected from a list of packages included in OSGeo4W when installing. Choose “Express Desktop Install” and accept all defaults except uncheck boxes for QGIS and OPEN GIS, when prompted to only install GDAL. Please see installation instructions at https://trac.osgeo.org/osgeo4w . The “Quick Start for OSGeo4W Users” section summarizes the installation instructions and provides links to download the installer for Windows operating system. Please note that the OSGeo4W installer requires internet access during the install process.

After installation, test by opening Command Prompt, type `gdalinfo --version` and hit enter. If no version number is returned, then GDAL needs be added to the system Path variable. To do this, open System Properties (enter "system environment variables" in the windows search box and selected "Edit the system environment variables") and select the box named "Environment Variables". In the "System Variables" box, click "Path" in the "Variable" column, then click "Edit" and then "New". Add (type or paste) the full path to the OSGeo4W binaries folder (e.g., C:\OSGeo4W64\bin). Click OK and OK again to close the Environment Variables window. Test again in a new Command Prompt window (it must be closed and re-opened to reset environment variables).

Mac operating system

On Mac, you will have to install the GDAL framework from an independent developer (a trusted source). Go to http://www.kyngchaos.com/software/frameworks and install the frameworks in this order:

1. GDAL 2.4 Complete
2. GSL framework v1.16-1
3. FreeType framework v2.4.12-1
4. PROJ framework v4.9.2-2
5. UnixImageIO framework v1.5.3
6. cairo framework v1.12.2-1

Because these are coming from an independent developer, you may see the following message when installing frameworks:

"[Name].pkg can't be opened because it is from an unidentified developer."

Once you see this message, go to your general security setting tab (System Preferences/Security & Privacy/General) and at the bottom you will see

"[Name].pkg was blocked from opening because it is from an unidentified developer."

Select "Open Anyway" and follow the command prompts. This will likely need to be done for each installation. Once you have completed step 6, open Terminal (Applications/Utilities/Terminal) and enter

```
export PATH=/Library/Frameworks/GDAL.framework/Programs:$PATH
```

To check that the installation was successful, enter `gdalinfo --version` at the terminal prompt.

Instructions for installing Rtools:

1. Point your browser to https://cran.r-project.org/bin/windows/Rtools/ . Select Rtools35.exe to download the file and then run the installer. Default options should be fine.

2. After installation on Windows operating system, Rtools will need to be added to the system Path variable. Open System Properties (enter "system environment variables" in the windows search box and selected "Edit the system environment variables") and select the box named "Environment Variables". In the "System Variables" box, click "Path" in the "Variable" column, then click "Edit" and then "New". Add (type or paste)

the full path to the Rtools binaries folder (e.g., C:\Rtools\bin). Click OK and OK again to close the Environment Variables window.

3. Check that Rtools was installed correctly using the R package ‘devtools’. In RStudio, install the R package ‘devtools’ using the R code below:

```
install.packages("devtools")
```

4. Use the ‘devtools’ function ‘find_rtools’ to check if rtools was installed corrected. Use the code below:

```
library(devtools)
devtools::find_rtools()
```

If the result is TRUE, you are done. Otherwise, a warning message will indicate that something went wrong. Double check that the Rtools\bin directory was added correctly to the system Path variable.

Instructions for installing the R package ‘glatos’:

1. In R, install the R package ‘remotes’ (if you haven't already); using the R code below.

```
install.packages("remotes")
```

2. Use the ‘remotes’ function ‘install_url’ to install the ‘glatos’ package and it's dependencies. Warning: this step will download 97 packages (dependencies) and it will take some time:

```
library(remotes)

install_url("https://gitlab.oceantrack.org/GreatLakes/glatos/repository/master/
archive.zip", build_opts = c("--no-resave-data", "--no-manual"))
```

3. Load the package and confirm that the version is correct (0.4.0 or newer; full-english-breakfast)

```
library(glatos)
```

4. Making animated videos with the glatos functions ‘make_frames’ or ‘make_video’ will require to external program FFMPEG. To check if FFMPEG is installed, use the ‘glatos’ function ‘check_dependencies’:

```
check_dependencies()
```

If FFMPEG is not found, procced to step 5.

5. The simplest way to install FFMPEG for this purpose is to install it directly into the glatos package library using the glatos function ‘install_ffmpeg’. Use the code below:

```
install_ffmpeg()
```

Note that this will need to be reinstalled (running the command above) each time a new version of the glatos package is installed. If a more permanent installation is desired, install from https://www.ffmpeg.org and add the path to the folder containing FFMPEG.exe to the system Path variable.

Instructions for installing the R package ‘VTrack’:

1. In R, install the R package ‘devtools’ (if you haven't already); using the R code below.

```
install.packages("devtools")
```

2. Use the ‘devtools’ function ‘install_github’ to install the ‘VTrack’ package from GitHub.

```
library(devtools)

devtools::install_github("rossdwyer/VTrack")
```

3. Load the package and confirm that it loads with no errors.

```
library(VTrack)
```