Installing GAT

Abigail Stamm, New York State Department of Health

Contents

About GAT

The Geographic Aggregation Tool (GAT) was created to simplify the process of geographic aggregation. GAT takes a set of user-defined parameters, aggregates based on these parameters, and outputs both the resulting shapefiles and documents to help the user assess the quality of the aggregation. GAT is installed and run as a package in R. Before you can use GAT, you will need to install R.

Installing R

You can download R at https://www.r-project.org/. Click on *download R* and select any site on the list. From there, choose your platform, then click on *install R for the first time.* Next, click on the download link to download R. If you need help installing R, click on *Installation and other instructions.*

Installing RStudio (optional)

After you have installed R, you can install RStudio, which is a wrapper that makes R more user-friendly and adds functionality, such as support for Markdown and Shiny and dedicated tabs for environments, tables, and images. RStudio **is not** required to run GAT.

You can download RStudio from https://rstudio.com/products/rstudio/download/. Under "RStudio Desktop Open Source License Free", click *Download.* Then click on the button, *Download RStudio for Windows.* After you have downloaded the file, install it as you would any other software program.

Installing GAT

GAT was compiled in R version 3.6 using RStudio and devtools. GAT has also been tested in R versions 3.4 and 3.5. R version 3.4 or greater is needed to install GAT.

Before installing GAT, install the necessary packages (or check that you have them installed). One way to do this is run the following code in your R console:

```
libs <- c("tcltk2", "rgeos", "maptools", "RColorBrewer", "classInt", "Hmisc",
          "foreign", "spdep", "plotKML", "sf", "sp", "lwgeom", "tibble",
          "graphics", "grDevices", "methods", "stats", "tcltk", "utils",
          "rgdal")
req <- unlist(lapply(libs, require, character.only = TRUE))
req <- libs[!req]
if (length(req) > 0) {
  for (i in 1:length(req)) {
    install.packages(req[i])
  }
  rm(i)
}
rm(req, libs)
```

Also check whether you have the XML package installed. If you do not or are not sure, this code will install it in Windows with R version 3.6.x:

```
pkg_link = "https://cran.r-project.org/bin/windows/contrib/4.0/XML_3.99-0.5.zip"
install.packages(pkg_link, type = "binary", repos = NULL)
```

Save the zip file for GAT. It will have a name like "gatpkg_<ver>.tar.gz" where <ver> will be the package version number.

In base R (not RStudio), click on "Packages" > "Install package(s) from local files...". Navigate to the gatpkg file and click "Open". Wait a few minutes while GAT installs.

If you prefer to install GAT through RStudio, click on "Tools" > "Install Packages...". In the "Install from:" drop-down, select "Package Archive File". A dialog should open immediately to select the "gatpkg_<ver>.tar.gz" file, but if it does not, click the "Browse..." button. Select the file. Click "Install". Wait a few minutes while GAT installs.

Using GAT

If you have not used GAT before, follow the tutorial vignette embedded in `gatpkg` to learn how GAT works. Access the tutorial by running the code,

```
browseVignettes("gatpkg")
```

Then click on the tutorial vignette HTML link. This will walk you through many of the options and settings available in GAT using shapefiles embedded in `gatpkg`.

The other vignettes embedded in the package include:

- **Technical Notes**: How each merge type works, compactness ratio, thinning
- **Setting up GAT**: A shorter version of this document
- **Change Log**: Differences between this version of GAT and the previous R script
- **Troubleshooting**: How to handle issues that arise in GAT
- **Shapefile specifications**: Requirements for a shapefile you plan to process with GAT

To run the default version of GAT, you only need one line of code,

```r
gatpkg::runGATprogram()
```

To learn about ways to customize GAT, check the technical notes or run the code,

```r
?gatpkg::runGATprogram
```

If you have used R before, `gatpkg` contains over 30 custom functions that can be modified or combined in different ways to meet your specific needs. Each of these functions has a corresponding help file that provides options and simple examples.

Uninstalling GAT

To uninstall `gatpkg`, run the code,

```r
remove.packages("gatpkg")
```