2024-03-14

geotargets: a working title

Eric R. Scott
University of Arizona, Communcations & Cyber Technologies Data Science

Signatories

Project team

Nicholas Tierney will serve as primary author and maintainer of `geotargets` during its development. Eric Scott (University of Arizona) will be a main contributor of code. Both anticipating using `geotargets` in ongoing and upcoming projects and have strong motivation to contribute to it's development.

Contributors

We already have contributions in the form of detailed reproducible examples in GitHub issues. In particular, Andrew Gene Brown at USDA-NRCS, has contributed code for dealing with reading and writing shapefiles, and choosing alternative filetypes for targets. Anthony North at Queensland Fire and Emergency Services has contributed code for using `geoarrow` as a backend for reading and writing targets. Dewey Dunnington at Voltron Data has also made suggestions on using `geoarrow`. Michael Sumner and Ben Raymond at Integrated Digital East Antarctica program, Australian Antarctic Division, have agreed to provide support and guidance in handling and managing geospatial data formats, in particular navigating GDAL.

Consulted

The idea for `geotargets` originated in a discussion posted to the discussion forum for `targets` where the author and maintainer of `targets` and other "Targetopia" packages showed strong support. Community contributions to `geotargets` via issues and comments on issues have been numerous despite the repository only having existed since the beginning of March 2024. Additionally, we plan to consult with geospatial experts throughout development to be sure we are addressing the most pressing issues and addressing them in ways that will fit with common geospatial analysis workflows in R.

The Problem

Geospatial computations in R are made possible by a group of R packages aimed at spatial and spatio-temporal data such as `terra`, `sf`, and `stars` (Hijmans, 2024; Pebesma & Bivand, 2023). Computations on spatial or spatio-temporal data can often be computationally intensive and slow when the underlying data is large (e.g. high-resolution global rasters). Depending on the data source and operation, this can range from taking seconds to days or weeks. Managing complex geospatial workflows can be confusing and re-running entire data pipelines is likely to be very time-consuming. The `targets` R package aims to aid with confusing and long-running workflows by automatically detecting dependencies among steps and only re-running steps that need to be re-run (Landau, 2021). This seems like a natural fit for complex geospatial workflows in R. However, geospatial packages like `terra` and `sf` don't work well with `targets` without extensive customization.

One notable difficulty is that `targets`, by default, saves R objects generated by computational steps, but R objects generated by the `terra` package may not actually contain the data itself but rather a C++ pointer to the data. When one of these `terra` objects is saved (e.g. as a .rds) and read back into R, it loses information about the data it represents and no longer works. To make these R objects portable and suitable for use with `targets` they need to be "marshaled" and "unmarshaled" requiring complicated code for a custom format.

A second obstacle is that often geospatial data is written in multiple files—for example, shapefiles, which are actually a collection of up to 12 files with different extensions. This limits compatibility with `targets` because the intermediate objects stored in a `targets` pipeline are required to be single files with no file extension.

Both of these challenges (and others) have been solved in bespoke ways for individual projects, but to date these solutions have not been formalized and distributed as an R package. Other packages exist as part of a "targetopia" that extend `targets` to work for specialized needs. For example stantargets for using targets with stan models. We hope that `geotargets` can join the "targetopia" and simplify geospatial data analysis with `targets`. We believe this will unlock a powerful workflow management tool for a large group of R users that have previously been unable (or unwilling) to use it because of these challenges.

The proposal

Overview

Our goal is to create a package that makes using `targets` for geospatial analysis in R as seamless as possible. To that end, `geotargets` will provide custom functions for defining geospatial targets that take care of translating and saving R objects for the user. In addition, we will create vignettes demonstrating how to use various geospatial R packages with `targets`. Where appropriate, we will identify contributions to existing R packages to make them easier to use with `targets` and `geotargets`.

Detail

In the `targets` package, analysis steps, or "targets", are defined with the `tar_target()` function. Targetopia packages provide additional `tar_*()` functions that extend `targets` by providing target archetypes for specialized workflows. The main contribution of `geotargets` will be a series of alternative `tar_*()` functions that create targets with pre-defined formats that take

care of the details of how these R objects are written out and read in by downstream targets. For example, to write a target that creates a raster using the `terra` package, one would use `geotargets::tar_terra_rast(name, command)`. `tar_terra_rast()` would provide a pre-defined format created with `targets::tar_format()` with functions for marshaling, writing, reading, and unmarshaling `terra SpatRaster` objects. In this case, marshaling/unmarshaling involves running `terra::wrap()` and `terra::unwrap()`, respectively, to make the R object "self-contained" rather than just containing a C++ pointer to the data. This is especially necessary for parallel computing with `targets` since `SpatRaster` objects don't work outside of the R session they were created in without `wrap()`ing them first.

As a minimum viable product, we hope to deliver an R package, hosted on GitHub, supporting raster and vector data objects from the `terra` and `sf` packages with custom target functions. Support for additional geospatial packages will be added based on feedback from the user community and through consultation with geospatial specialists. In initial development we will choose sensible defaults for what file types targets will be stored as (e.g. GeoTIFF for raster data). In the future we will develop a `filetype` argument for each `tar_*` function, since there are many options for how geospatial data can be stored on disk by these packages. For example, "netCDF", "HEIF", and "BMP", and 161 other options listed in the GDAL raster driver. This will offer flexibility in light of trade-offs between file size, read/write speed, and dependency requirements similar to the existing options for how objects are stored by the `targets` package (i.e. default '.rds' with options for faster/smaller file types).

Project plan

Start-up phase

We have already created a repository on GitHub that uses GitHub actions to run package checks. In the start-up phase, we will focus on making design decisions about what the package will offer and research and discuss what r-spatial packages and object types will we support. Answering these questions will lay the groundwork for efficient collaborative development of the package.

Technical delivery

Our goal is to deliver a package that allows users to use the `targets` package with various r-spatial packages (`terra`, `sf`, `stars`, etc.) with as little friction as possible.

Milestone 1: July 31

- Basic package with functionality for `terra SpatRaster` and `SpatVector` objects
- Well documented functions with high test coverage ensuring a solid start to the project
- Creation of a `pkgdown` website for the package hosted on GitHub pages

Milestone 2: September 30

- Add support for objects from a second r-spatial package such as `sf`
- Benchmarking of various file type options for storing targets including file size and read and write speed. Our findings will be published as an article on the `geotargets` website.

Milestone 3: November 30

- Add support for objects from a third r-spatial package.
- Prepare for submission to rOpenSci

Other aspects

Throughout the project we will seek feedback from users on social media (Mastodon and Twitter) and on the `targets` discussion forum. Each milestone will coincide roughly with a release on GitHub and a short blog post on https://www.njtierney.com/. We intend to submit the package to rOpenSci for software review, and subsequently submit for publication to the Journal of Open Source Software.

Requirements

People

Processes

Tools & Tech

Funding

Summary

Success

Definition of done

Measuring success

Future work

Key risks

References

Hijmans, R. J. (2024). Terra: Spatial data analysis. https://CRAN.R-project.org/package=terra

Landau, W. (2021). The targets r package: A dynamic make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. *Journal of Open Source Software*, *6*(57), 2959. https://doi.org/10.21105/joss.02959

Pebesma, E., & Bivand, R. (2023). Spatial Data Science: With applications in R, 352. https://doi.org/10.1201/9780429459016