Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling

Daniel Nüst

2020-07-13

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://osf.io/ztc7m/. To cite the report use

> *Nüst, D. (2020, June 5). Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling. https://doi.org/10.17605/OSF.IO/ZTC7M*

Reviewed paper

> *Hojati, Majid and Robertson, Colin: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling. AGILE GiScience Ser., 1, 6. https://doi.org/10.5194/agile-giss-1-6-2020, 2020.*

Summary

The paper code and a sample dataset were published in a public GitHub repository. The repository includes a working Binder link. The workflow in the repository could be executed after fixing two small bugs. However, the workflow does *not* create all of the figures in the paper, because only the test data is published with the repository while further data sources are only documented. While *the reproduction was only partially successful*, the authors demonstrate good concern for the reproducibility of their work. The reviewer notes therefore include some comments for possible improvements, some of which were already adopted by the authors after sending them a first draft of this report.

Reproducibility reviewer notes

Reproduction

The paper contains an a Data and Software Availability section. All scientific reviewers took note of the repository, one attempted to run the code (but stopped after first errors), and another browsed the contents of the repository. I turned straight to the section after reading the abstract and skimming the remainder of the manuscript. The section describes the limitations of the workflow and points to a test dataset and code in a public GitHub repository at https://github.com/am2222/AGILECA, which I forked to https://github.com/reproducible-agile/AGILECA. The workflow also has a documentation website at https://am2222.github.io/AGILECA/ but it only mirrors the information from the paper. The Zenodo repository, http://doi.org/10.5281/zenodo.3586312, seems to contain an outdated release of the code not matching the GitHub project.

The documentation included a Binder link, which I ran: https://mybinder.org/v2/gh/am2222/AGILECA.git/master The reviewed revision is `45b7b8768634693324f91cc789341da447686c80`.

In the first code chunk, line 69 gives an error:

```
Error: stat_sf requires the following missing aesthetics: geometry
```

I could fix this by explicitly setting the geometry aesthetic in the `plotResult()` and `finalPlot()` functions:

```
geom_sf(data=sf_df, aes(fill = step, geometry = wkt), [..]
```

In line 110, I got the following error:

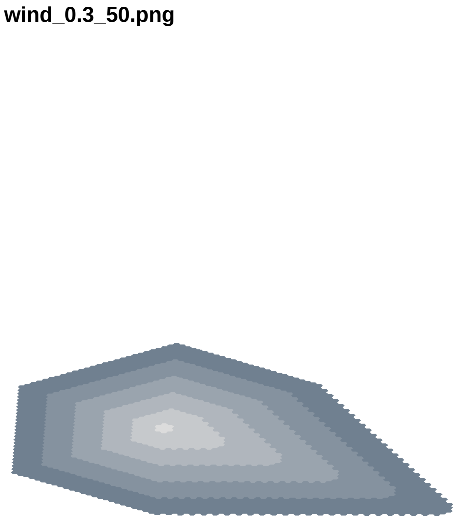
```
Error: Join columns must be unique. x Problem at position 2.
```

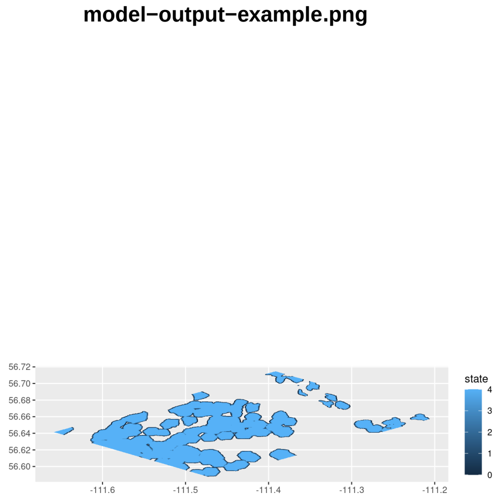
This could be fixed by applying the change in the function `testWind()` also to `testLanduse()` in line 356 of `model_test_cases.R`:

```
left_join(lookup,by=c("dggid"))%>%
```

With the same fixes, I was also able to render the full document locally. The rending on MyBinder then also works, but only chunk by chunk - it seems rendering the full document exceed the timeout of the free service: https://mybinder.org/v2/gh/reproducible-agile/AGILECA/master?urlpath=%2Frstudio

The created HTML file `run-model.html` is included in the reproduction project on OSF and includes several plots matching parts of Fig 6. of the article and the example figures in the repository README.





Conclusion

A clear connection which code produces which figure was not provided. Each of the plots took at least few minutes to render. A time to be expected to run the workflow was not provided.

Core figures of the articles, such as Fig. 6 (c), Fig. 7 or Fig. 8, are *not* created with the published code but rely on additional data. These datasets are described in detail in Table 1 of the paper, but due to the lack of step-by-step instructions and the complexity to retrieve the input data from different sources (including ones requiring accounts), a full reproduction was not attempted. The following data source links were checked and match the description in the paper, however, none of these are *direct* links to the datasets that were supposedly used in the presented work.

- https://doi.org/10.5067/FIRMS/VIIRS/VNP14IMGT.NRT.001
- https://doi.org/10.24381/cds.e2161bac
- https://open.canada.ca/data/en/dataset/7f245e4d-76c2-4caa-951a-45d1d2051333
- https://www.usgs.gov/landsat

Comments to the authors

The given repository is a good attempt at trying to make your workflow partially reproducible. When I shared the preliminary report with the authors, they reacted promptly and have fixed some of the comments below. See this link for a comparison of the current state with the originally reviewed revision `ab1a67c9ad1734114ca667cb9e6d9933774fbae3`.

I have the following concrete recommendations:

- Most importantly, provide detailed instructions how to access data files from Table 1, ideally using scripts you provide, and include datasets where the license allows in your repository (unless to large - then document much more detailed how to recreate the inputs, i.e., the specific part of the referenced datasets)
- create a better connection between the figures in the article and your source code. Ideally, use the same numbering in your Rmd file as in the paper
- Zenodo deposit is not up to date, suggest to keep GitHub and Zenodo in sync by making a new release
- Your repository README could describe the steps to reproduce the workflow a bit better and explain, what a person can expect (i.e., not getting the figures from the paper because the data is not included)
- [Fixed] Better connect the figures in the Rmd file with the paper
- Give the code chunks in R Markdown useful names
- You can combine the different figures into one plot with `patchwork` so the code creates the complete Figure 6
- [Fixed] In `install.R`, if installing from GitHub, use `ref =` for better reproducibility (installing a specific commit or tag, not from any branch)
- [Fixed] Artifacts from basing the analysis on https://github.com/nuest/sensebox-binder/ (GIF, `sensebox-analysis.ipynb` file) I would have appreciated a note in the README that you used that project as a template
- `run-model.Rmd` has a link to https://doi.org/10.5281/zenodo.1135140, which is the wrong deposit
- [Fixed] Binder link does not work
- You can use the `urlpath` parameter for Binder to directly open RStudio: https://mybinder.org/v2/gh/am2222/AGILECA/master?urlpath=%2Frstudio, see MyBinder docs
- Do not do `setwd(here())` - just use the `here()` function when loading files
- Use a code formatter for better readability

Problems during first review

1. Stepping through the code chunks in `run-model.Rmd`, the code failed in line 68, with the following error:

   ```
   Error in geom_sf(data = sf_df, aes(fill = step), lwd = 0, color = NA) :
   ```

   I inspected the `sessionInfo()` and assume this problem is because the Binder used an outdated version of R with an old MRAN snapshot (`https://mran.microsoft.com/snapshot/2017-11-30`). *This problem was fixed by the authors when updating the used R version from `3.4.2` to `3.6.2`.*

2. Unfortunately system specific path delimiters were used in line 141, so the code did not run. I could fix this by changing the line to

   ```
   fire <- readOGR(here("data", "DL_FIRE_M6_62518", "fire_archive_M6_62518.shp"))
   ```

   which takes advantage of the features of the `here::here()` function. *The authors adopted this suggestion.*

3. The file `nghbs.txt` is missing, so I instead loaded `nghb_test.txt`, which seems to be correct as the *change was applied by the authors, too.*