How to combine R with AUTHOREA to collaboratively write reproducible scientific papers

Tobias Roth[1]

[1]University of Basel

June 22, 2019

Abstract

THIS IS WORK IN PROGRESS.

Introduction

Recently, we submitted our first manuscript that was entirely written in Rmarkdown. See (Roth et al., 2018) for the manuscript. The code that was used to produce this manuscript can be downloaded from our GitHub repository . Rmarkdown in connection with the version control provided by git / GitHub is a great environment to develop the statistical analyses from the first ideas to the final results. It is also a great way to provide reproducible data analyses to reviewers and readers. However, we felt that the system was not perfectly adapted to a smooth writing process, which includes the writing of the first draft of the manuscript in parallel with developing the statistical analyses as well as the process of developing the final MS from the first draft in collaboration among all the authors (we were eight). In our point of view our workflow had the following shortcomings:

- It is not an environment that makes it easy for co-authors to quickly correct a spelling mistake or to better formulate a bumpy sentence. So I ended up with receiving a lot of mails with comments pointing to some lines within the pdf of the manuscript that should be improved. It took quite an effort to find the corresponding line in the Rmarkdown files to make the corrections and improvements.
- Most coauthors were not comfortable with writing directly into my Rmarkdown files because they were afraid of corrupting the R-code.
- When I worked on the text I ended up switching back and forth between the formatted version of the manuscript (pdf in our case) to read the text and to the Rmarkdown files to make the corrections.
- Handling the references in Rmarkdown-files is a bit clumsy especially if you have a lot of them.

So after this experience I was hoping to slightly improve the writing part of the workflow for the next project. My wish list included the following:

- A workflow that is compatible with R and GitHub but that is also compatible to work with co-authors that won't get passed Word-like editing of text (see also this thread)
- The system should be able to digest figures and tables (preferably in Latex) that are produced from R.
- It should also be possible to have parts of the manuscript which contain of text and numbers that were produced by inline R-code. This is especially handy in the results parts to add some summary statistics without the need to copy past.

- An easy way to search for references on the web and including them directly into the text as well as into the .bib file.

When searching on the web to improve things I came across several sources that mentioned Authorea . Authorea is a collaborative writing platform for researchers. Authorea has some features that sounded promising to me:

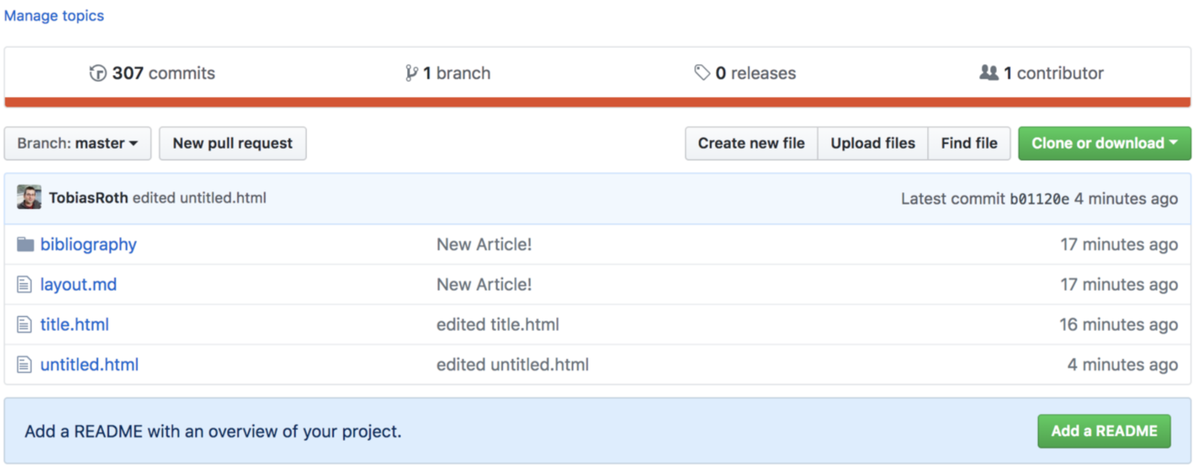
- Authorea can easily be linked with an existing GitHub account.
- Authorea's instant search to locate a reference by author, keyword, or DOI makes it very easy to add references.
- Authorea not only understands (rich) text, markdown, html and latex, these components can also be mixed in the same manuscript which seemed quite unique to me.
- Authorea splits the manuscript into blocks (title, abstract, textblocks, figures, tables) and has a single file (layout.md) that defines how these blocks are grouped together to from the entire manuscript.

So I decided to give it a try. This document is the result of this try. The GitHub repository that is linked to this article can be found at https://github.com/TobiasRoth/RtoAuthorea.

Connecting an Authorea document to own GitHub repository

New repository from scratch

- Add new document to AUTHOREA and just write the name of yourrepository as the title of the document. Then setup automatic GitHub integration.
- Go to your GitHub account and the new repository should already be there.



TobiasRoth edited untitled.html — Latest commit b01120e 4 minutes ago

bibliography	New Article!	17 minutes ago
layout.md	New Article!	17 minutes ago
title.html	edited title.html	16 minutes ago
untitled.html	edited untitled.html	4 minutes ago

Add a README with an overview of your project.

Figure 1: This is how the new document looked on the GitHub-Account.

Adding AUTHOREA to an existing repository

When you aim to add AUTHOREA to an existing directory you can add a new document in AUTHOREA using the name of the repository as the title of the document. Then setup automatic GitHub integration and the AUTHOREA files will be added as a separate branch in the corresponding repository on GitHub.

Just make sure that the AUTHOREA files wont conflict with existing files in the repository and merge to AUTHOREA-branch into the master branch with the following command.

```
git merge origin use-ts-bot --allow-unrelated-histories
```

Adding content

The easiest way to add content is to use the AUTHOREA interface. Add a placeholder, tidy up the files in Github and then add R-scripts to change part of the content.

Add a figure

Whenever you have an R-Script that is producing a figure that you like to include into the manuscript, just save the figure at a place of your choice. Go to authorea and add this figure into the manuscript using the `Insert/Figure` button. Authorea will then add all the information needed including the figure into a separate folder within the `figures` folder. You can now change your R-script to overwrite the figure that was now added by authorea. Thus, whenever you re-run the R-script the newly produced figure will be automatically included into the manuscript as soon as you commit your changes to the GitHub repository.

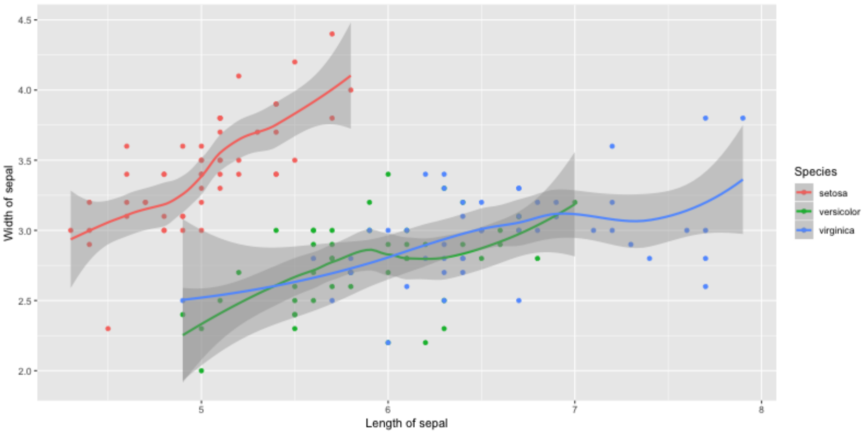


Figure 2: This figure was produced by this R-Script. It shows a scatter plot of sepal length against sepal with of the three iris species.

Adding a Latex table

Be aware that complex latex are perfect to show online and to export to PDFs or Latex, but they are unlikely to show well when exporting to word. To include a latex table we first include a latex junk in the AUTHOREA document, then we tide things up, which means that we give a sensible name to the new file, put it in the appropriate directory and adapt *layout.md* accordingly. Then we add an R file that produces the latex code of the table and write it into the *.tex* -File. The R-code to produce Table 1 can be inspected here. When clicking on the table itself the latex code that was produced by the R script is shown. When collaborates change this latex code (e..g to change the caption) you need to make sure to manually include the changes into the R script that is producing the table. This can easily be done with

Table 1: This is a summary table of the famous iris data set.

Species	n	average PL	SD PL	min PL
setosa	50	5.006	0.3524897	0.3524897
versicolor	50	5.936	0.5161711	0.5161711
virginica	50	6.588	0.6358796	0.6358796

R-code

the `diff` comparison that is implemented in RStudio: when you run the R-script to produce the Latex code for the table just hit the `diff` button in the `git` tab. The differences between the old version and new version of the .tex file will be shown and you may see some changes that you should manually copy into the R-script.

Questions to Authorea

How to solve merging conflicts?

At some stage I run into merging conflicts. This was because I added a figure within an existing text bloc using the Authorea interface . Authorea then needs to splits the text bloc into two and adds a new html file to the repository that contains the second part of the original text bloc. In the background, Authorea of course integrates the new html file also into`layout.md` to make sure that the new text bloc is included into the document. I wasn't aware of that and changed `layout.rm` from R at the same time, which obviously led to a merging conflict. In case of a merging conflict Authorea pushes to a separate branch (by default this branch is labeled as *authorea*, but you can change it in the deploy key settings). To fix the problem I created a new pull request from my GitHub repository and merged the *authorea* branch back into the master branch. This worked well and everything seemed to be fine again. However, I then started to change the figure from R as described above. The new figure turned up in the `Data` folder of the Authorea document as expected but it did not change in the document itself. I suspected that this was an issue related to the cache of the browser, but simply reloading the html-page did not help. Furthermore, I soon ran into the next merging conflict that I did not really understood. I was not able to really understand the issue and to definitively solve it using pull request. Thus I decided to completely delete the document on Authorea and connected theGitHub repository to a new and empty Authorea document as explained above. I just opened the `layout.md` file from Authorea (from the `Data` folder) and pasted the content from the old `layout.md` file. After having done so the document reappeared as it used to be and everything worked fine again (including changing the figure from R). I think the short question is: *How to properly solve a merging conflict?*

Why Authorea stores a copy of a figure?

When adding a figure using the Authorea interface, Authorea seems to save the figure two times (one is labeled as the original). What is the use of it? This question is emerging when I aim to change a figure from R. Which figure do I need to change? I suspect that either figure could be used depending on how the document is viewed / exported (html vs. pdf vs. docx)? Is there a figure format that should be preferred? I personally would prefer to use .pdf or .png?

When do I need to manually pull or push from Github?

It is rather easy to push and pull from the Authorea interface. You just need to add `/git_functions` at the end of the url and you get access to the two Git functions. When changing things from R it seems sometimes necessary to manually pull from or push to GitHub. However, I do not yet really understand when this is the case. Maybe I was also a bit misguided by the merging conflict described above and it is usually not necessary?

Discussion

Her we document our first experience with using Authorea in connection with R and GitHub. We aimed to better understand the behavior of Authorea and asked whether it could make sense to include Authorea into the workflow of conducting a data analyses project that should be finally published in a scientific manuscript. In short, we are very positive that integrating Authorea into our current workflow with R and GitHub could help to considerably improve the writing experience . We are also confident that it is in line with the emerging topic of reproducible research. So this document can be considered as a proof of concept.

In principle connecting the R/GitHub workflow with Authorea works quite well out of the box, but obviously the handling and tidy up of the different files is quite clumsy. We like to end up this document with starting a discussion about thing that might be developed (for instance as an R-package) that would help better incorporate Authorea into our R/GitHub workflow.

R-Function to automatically include a figure

Let's assume that I produced a plot in R that I would like to include into the Authorea document. It would be nice to have an R-Functions that would do this for me. Some

Add Authorea to the Rmarkdown workflow

This is more a goal in the long run. I suspect it might be possible to implement a new output format for R markdown. So I could write a normal Rmakrdown file and in the header of the document I would specify something like `output: authorea_document`. When rendering the document figures and tables would be automatically added into the right place as described above. Normal text (including R junks) might be rendered to markdown and included as markdown into the document. I think the most difficult issue in this respect is how to feed the changes made in the Authorea document back into the Rmarkdown files. I suspect this might be possible using the following workflow:

1. When rendering the Rmarkdown to markdown execute the R-junks as usual but additionally add the R-juncts as markdown comment into the markdown file. Thus the R-code is still there but will not be shown in the Authorea document.
2. Let the co-authors make changes in the text using the Authorea interface.
3. Make a function that takes the markdown file (that may contain edits from the coauthors made using the Authorea interface) and re-builds the Rmarkdown file (it basically needs to remove the numbers that were originally included by running the R-juncts and uncomment the R-junks.
4. Maybe the function of 3. could be automatically run whenever one is using the *push* button in RStudio to get the changes from the remote git repository (GitHub).

References

Roth T., Kohli L., Bühler C., Rihm B., Meuli RG., Meier R., Amrhein V. 2018. Species turnover reveals hidden effects of decreasing nitrogen deposition in mountain hay meadows. DOI: 10.7287/peerj.preprints.27230.