Reproducible science: Module 9

Workflow for reproducible work: Project

Gbadamassi G.O. Dossa

Xishuangbanna Tropical Botanical Garden, XTBG-CAS

2021/10/22 (updated: 2022-06-29)

Acknowledgements

The content of this module are based on materials from:

Rob Schick's materials

Brainstorm with class

- What are critical elements needed?
- What has been a problem?
- What do you consider a need?

Consistent Themes

- Working with collaborators
- Consistent uncluttered file structure
- Ability to re-enter the project and quickly reorient
- Knowing what to version control
- Knowing version control
- Testing new code without breaking original
- Re-running things efficiently

What is a *project*

A project is a well organized folder or directory.

A project must contain all data related to it. Thus, we must have:

- all data the project is based upon;
- all the code (for cleaning and carrying out data analysis);
- all necessary text that explains how the code talks to the data to render results;
- any final reports derived from the project management;
- any license entailed to the project;
- a clear record of how the project has evolved through time (like lab notes, measurement protocol etc.)

A good Enough starting point

“Computing workflows need to follow the same practices as lab projects and notebooks, with organized data, documented steps, and the project structured for reproducibility, but researchers new to computing often don’t know where to start.” ---Wilson et al. (2017)

Six core tenets of a good enough practice

Data Management
Save the raw data
Back them up in >1 location
Create the data you wish to see in the world
Create analysis-friendly data
Record all the steps used to process data (use version controlled code for this)
Anticipate needing multiple tables; so use unique IDs
Submit data to a reputable DOI-issuing repository so others can access and cite

Six core tenets of a good enough practice 2

Software
Place a brief explanatory comment at the start of every program
Decompose programs into functions
Be ruthless about eliminating duplication
Search for libraries that do what you want to do
Test them before relying on them
Give functions and variables meaningful names
Make dependencies and requirements explicit
Do not comment and uncomment sections of code to control a program's behavior (we'll come back to this)
Provide a simple example or test data set
Submit code to a reputable DOI-issuing repository

Six core tenets of a good enough practice 3

Project Organization
Put each project in its own directory, which is named after the project
Put text documents associated with the project in the `doc` directory
Put raw data and metadata in a `data` directory and files generated during cleanup and analysis in a `results` directory
Put project source code in the `src` directory
Put external scripts or compiled programs in the `bin` directory
Name all files to reflect their content or function

Six core tenets of a good enough practice 4

Keeping Track of Changes
Back up almost everything created by a human being as soon as it is created
Keep changes small
Share changes frequently
Create, maintain, and use a checklist for saving and sharing changes to a project
Add a file called `CHANGELOG.txt` to the project's `doc` subfolder
Use a version control system

Six core tenets of a good enough practice 5

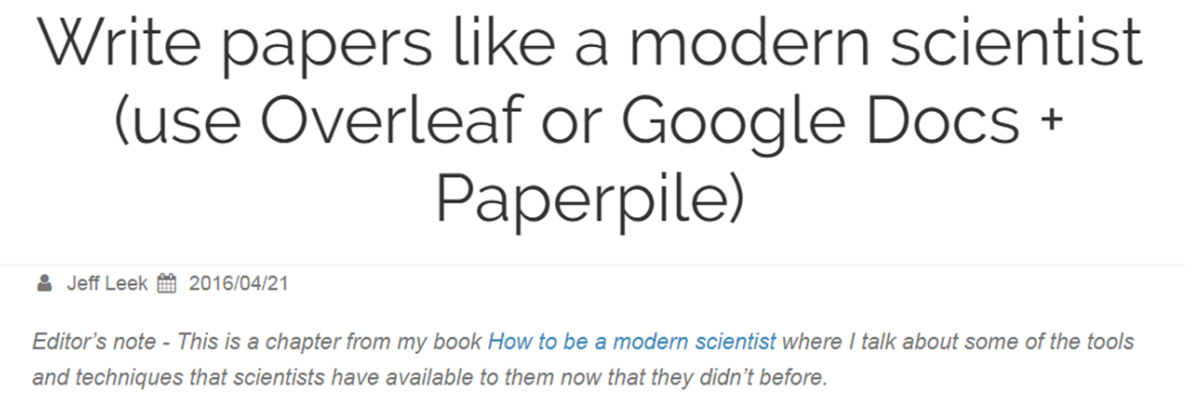
Keeping Track of Changes
Back up almost everything created by a human being as soon as it is created
Keep changes small
Share changes frequently
Create, maintain, and use a checklist for saving and sharing changes to a project
Add a file called `CHANGELOG.txt` to the project's `doc` subfolder
Use a version control system 🙂

Six core tenets of a good enough practice 6

Manuscripts
Don’t use MS-Word, Pages, Libre Office, etc.
Write manuscripts using online tools with rich formatting, change tracking, and reference management
Use a version control system

Six core tenets of a good enough practice 7

Getting Collaborators on Board - ymmv



Write papers like a modern scientist (use Overleaf or Google Docs + Paperpile)

Jeff Leek 2016/04/21

Editor's note - This is a chapter from my book How to be a modern scientist where I talk about some of the tools and techniques that scientists have available to them now that they didn't before.

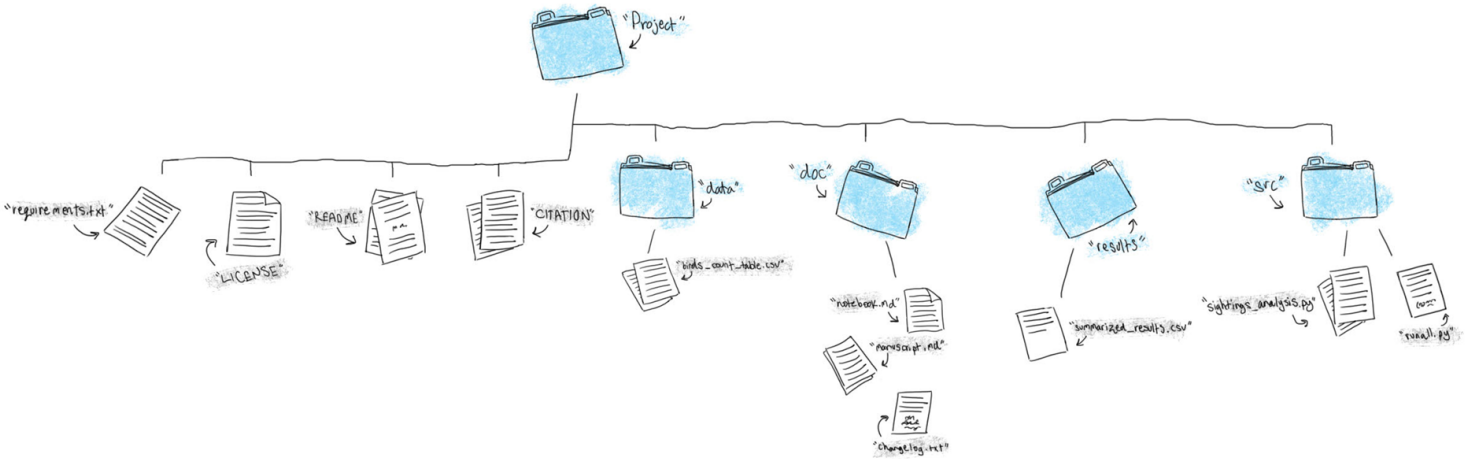
See magnificent tips from from Jeff leek

Six core tenets of a good enough practice 8

Best Practices
Data Management
Software
Collaboration
Project Organization
Keeping Track of Changes
Manuscripts

Six core tenets of a good enough practice 9

Below is graphically how the structure of a good enough project should look like



Drawing by Stella Schick

Drawing by Stella Schick

Let's Build and Populate a Project

- Navigate to a directory
- Make a folder for your project named whatever you want
- cd into the project folder
- Start making folders with mkdir
- Create 4 files using a text editor, or the touch command
    - CITATION;
    - README;
    - LICENSE;
    - requirements.txt

Have you succeeded?

Here is how it should look like on your git bash. Does yours look similar?

```
rob@rob-win7-guest MINGW64 /e/rob/Documents/business/2017_ESA/tallgrass
$ ls -lt
total 0
-rw-r--r-- 1 rob 197121 0 Aug  3 14:53 requirements.txt
-rw-r--r-- 1 rob 197121 0 Aug  3 14:53 LICENSE
-rw-r--r-- 1 rob 197121 0 Aug  3 14:53 README
-rw-r--r-- 1 rob 197121 0 Aug  3 14:53 CITATION
drwxr-xr-x 1 rob 197121 0 Aug  3 14:52 src/
drwxr-xr-x 1 rob 197121 0 Aug  3 14:52 results/
drwxr-xr-x 1 rob 197121 0 Aug  3 14:52 doc/
drwxr-xr-x 1 rob 197121 0 Aug  3 14:51 data/
```

What goes in each part? – Data Folder

- Raw data
  - Consider locking the file, or making read-only
  - Never hand edit the file!
- Metadata
  - I have one .csv file from BMMRO

What goes in each part? – src Folder

- Two types of scripts
  - Individual analytical blocks
    - reshapeData.R
    - runRegression.R
- Controller file, e.g. runAll.R
- runAll.R may contain:
  - source(reshapeData.R)
  - source(runRegression.R)
  - source(plotAnalysis.R)

And can be called from a command prompt with: R CMD BATCH –vanilla runAll.R runAll.rout &

What goes in each part? – results Folder

- Any “generated” result:
  - Intermediate results
    - Cleaned data
    - Simulated data
  - Final Results
    - Figures
    - Tables
- For most projects, you’ll likely have sub-directories:
- Results
  - CleanedData
  - Figures
  - Tables

What goes in each part? - doc Folder

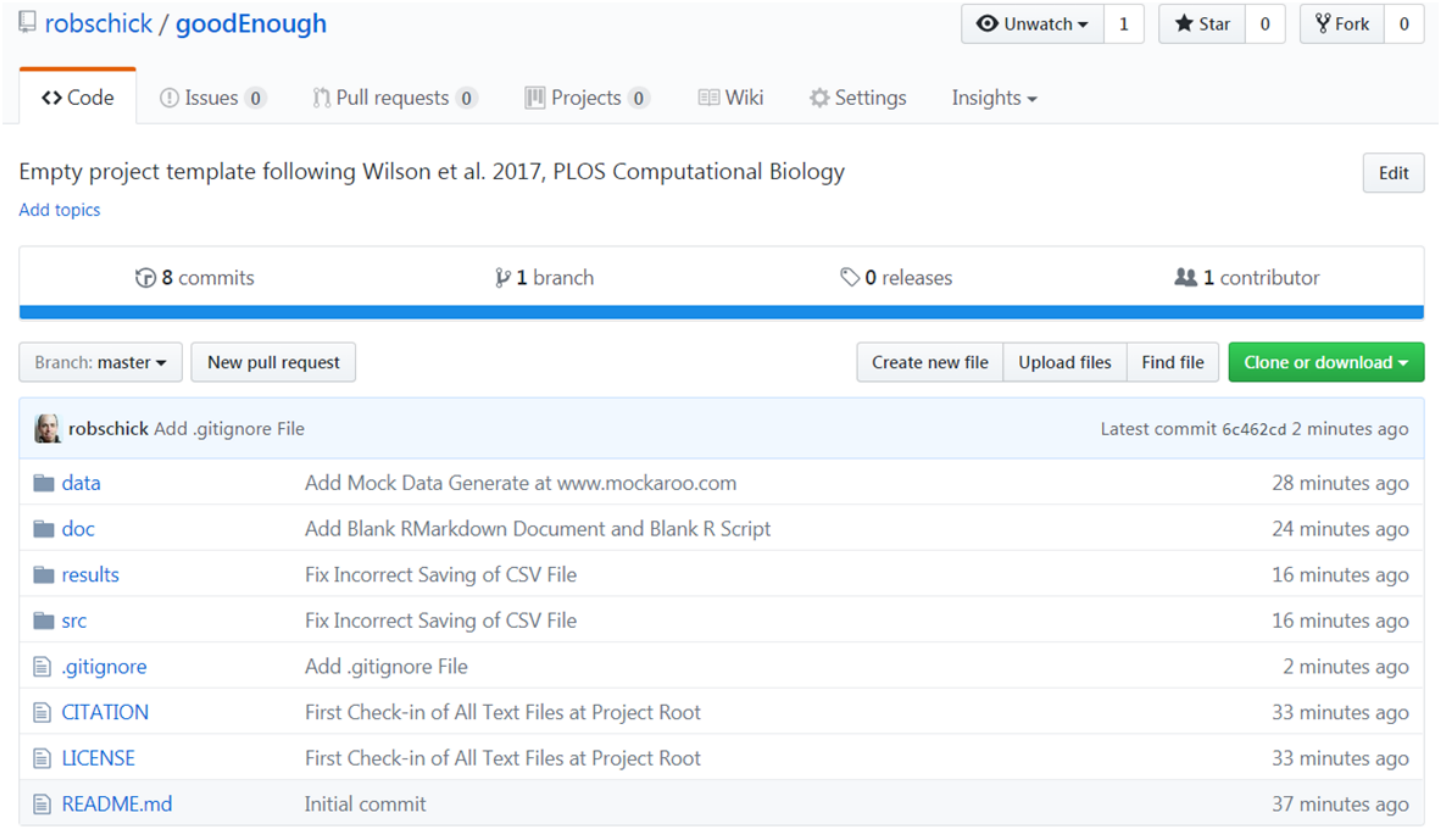
- Any text based documents:
- If manually versioning, a changelog.txt file
- projectNotebook.md (maybe Notepad or evernote…)
- Manuscript (if using a version control, otherwise Google Docs)

What goes in each file?

- The README: file provides an overview of the project as a whole
  - Project's title
  - Brief description
  - Up-to-date contact information
  - An example of how to run the code
  - How people can engage with the project
  - If you are looking for contributors
- CITATION: Explains how to reference the project:
  - DOIs for code and data
  - DOI for project if using OSF
  - Manuscript Example here
- LICENSE Explains the licensing, e.g. CC-BY, CC-ND, etc.

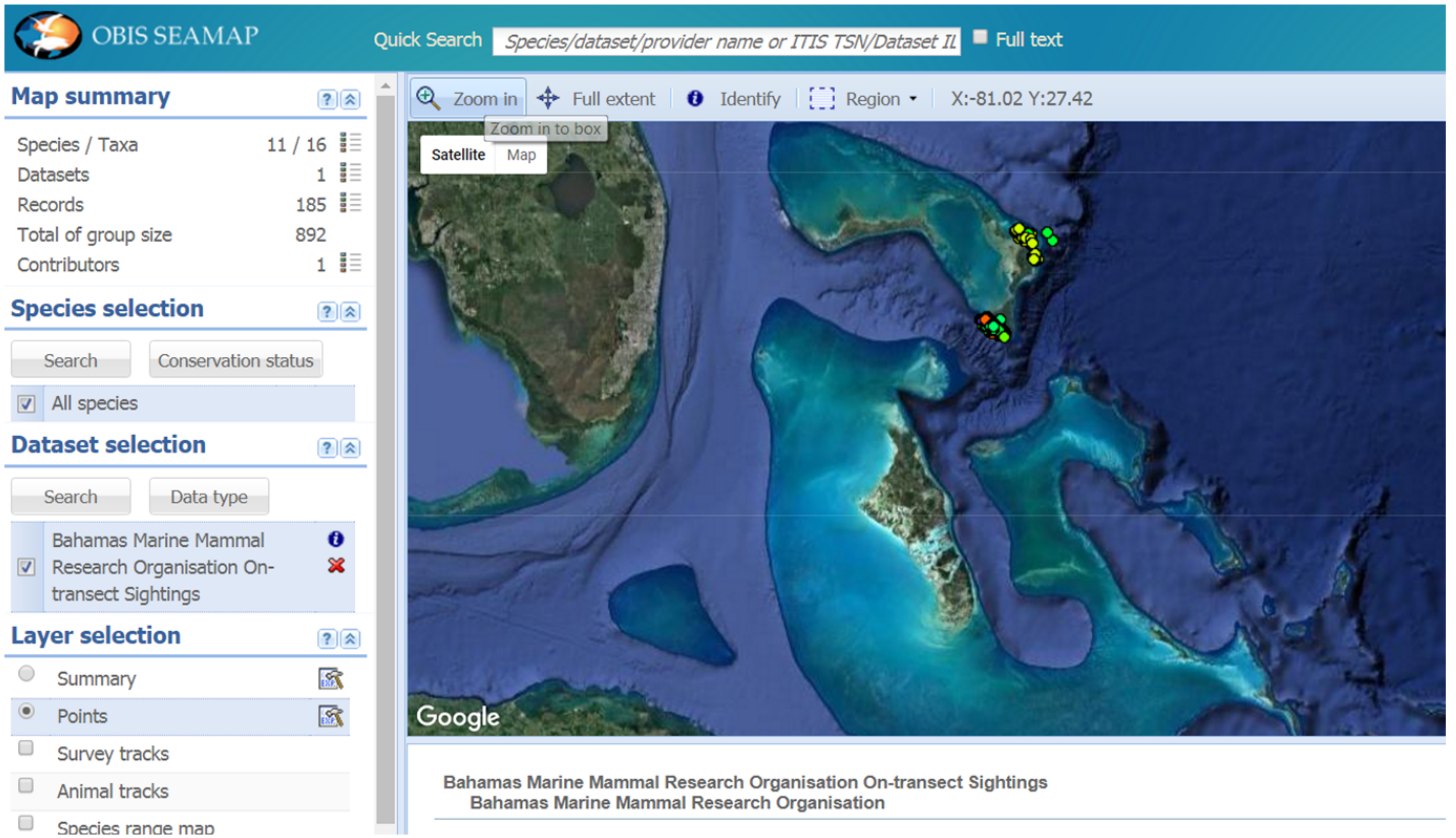
Get a copy of a good enough practice

You can get a template of a good enough Project structure as repository from Rob Schick Github



Good enough project workflow practice

Bahamas Marine Mammal Research Organization (BMMRO) data



Thank you for listening!

Any questions now or email me at **dossa@xtbg.org.cn**

Slides created via the R package **xaringan**.

The chakra comes from remark.js, **knitr**, and R Markdown.