Software Best Practices for Reproducible Open Science

PyHC Summer School – May 20th, 2024

Alex Koufos (he/him)

Research Software Engineer - Stanford University

Steering Committee Member - US-RSE

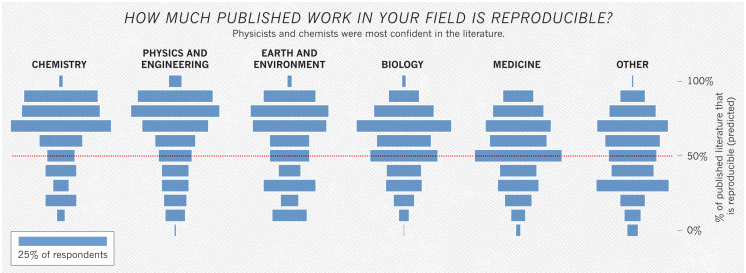
Overview

- Intro
  - A Discussion on Reproducibility
- Software Best Practices
  - Version Control
  - Code Reviews
  - Documentation
- Conclusion
  - Further Practices
  - Some Resources

Poster version

Intro: Expectation for Reproducibility

Studies have shown that confidence in reproducibility is a potential concern[1][2]



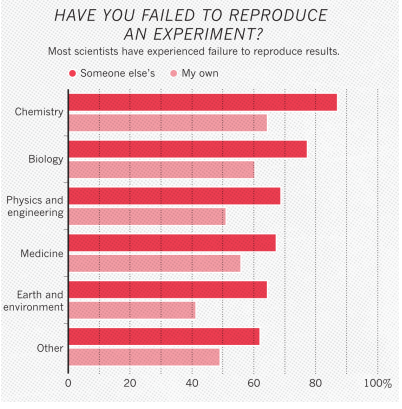
Science Field Based Confidence in Published Work[1]

[1] 1,500 scientists lift the lid on reproducibility - Baker 2016

[2] Reproducibility of Scientific Results - Fidler and Wilcox 2021

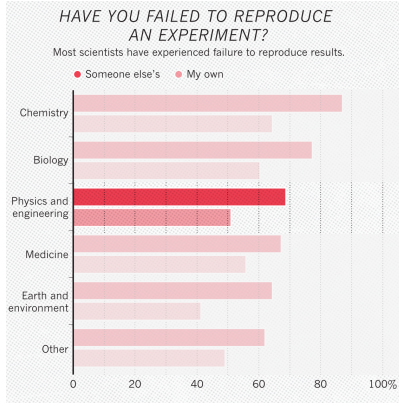
Intro: A "Reproducibility Crisis"

- The reality doesn't meet those expectations



Intro: A "Reproducibility Crisis"

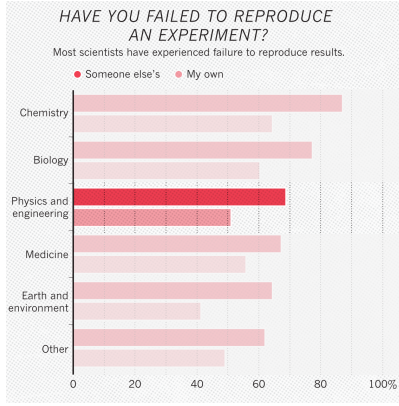
- The reality doesn't meet those expectations
- In Physics/Engineering, nearly 70% could not reproduce other's work



[1] 1,500 scientists lift the lid on reproducibility - Baker 2016

Intro: A "Reproducibility Crisis"

- The reality doesn't meet those expectations
- In Physics/Engineering, nearly 70% could not reproduce other's work
- Over 50% could not reproduce their own work



[1] 1,500 scientists lift the lid on reproducibility - Baker 2016

Intro: What is Reproducibility

The National Academies of Sciences, Engineering, and Medicine's 2019 report, defines[1]:

Reproducibility (i.e. computational reproducibility)

> as obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis

Replicability

> as obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data

[1] Reproducibility and Replicability in Science - National Academies Report 2019

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**
- Enhanced **collaboration**

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**
- Enhanced **collaboration**
- Boosted **clarity**

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**
- Enhanced **collaboration**
- Boosted **clarity**
- Greater **transferability** of data and programs

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**
- Enhanced **collaboration**
- Boosted **clarity**
- Greater **transferability** of data and programs
- **Transparency** of results and methods

Best Practices: Overview

Version control, **code reviews**, and **documentation** can help with the goals of:

- Better **reproducibility** and **robustness** of software
- Improved **scalability**
- Enhanced **collaboration**
- Boosted **clarity**
- Greater **transferability** of data and programs
- **Transparency** of results and methods
- Higher **reusability** of code

Best Practice: Version Control

What is version control?

- A history of your software

Why use version control?

- Touches **reproducibility**, **robustness**, **scalability**, **collaboration**, and **transparency**

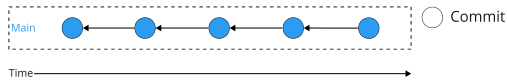
Let's take a look at Git

How not to version control[1]

[1] PhD Comics - Cham 1997

Best Practice: Version Control (git)

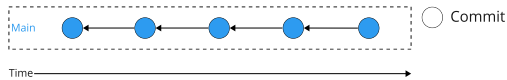
- Each circle is a ”commit” or snapshot of the repo
- Arrows represent the connection between commits
- Allows use to easily compare changes



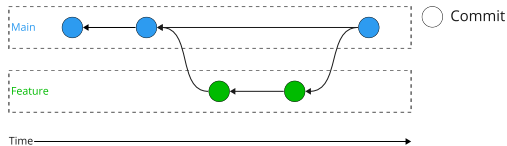
A simple Git history of the ”main” branch

Best Practice: Version Control (git)

- Each circle is a ”commit” or snapshot of the repo
- Arrows represent the connection between commits
- Allows use to easily compare changes



A simple Git history of the ”main” branch



A Git repo using a branching model known as GitHub Flow

Best Practice: Code Reviews

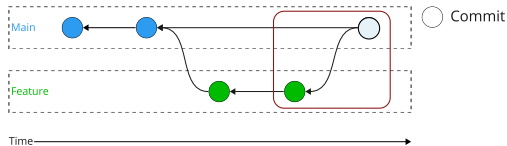
What are code reviews?

- Process to review changes

Why use code reviews?

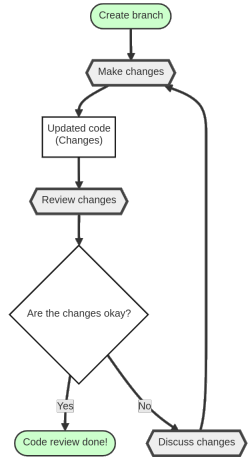
- Touches **reproducibility**, **robustness**, **scalability**, **collaboration**, **clarity**, and **transferability**

In general, the process is three or four steps



Best Practice: Code Reviews (Cont.)

1. Update your codes (i.e. make changes)
2. Compare differences between initial snapshot (i.e. review changes)
3. Make suggestions and comments (i.e. discuss changes)
4. Merge changes into stable code.



Flow diagram of a review process

Best Practice: Documentation

What is documentation?

- The process of explaining the functionality and use of your software via some text-based system.

Why use documentation?

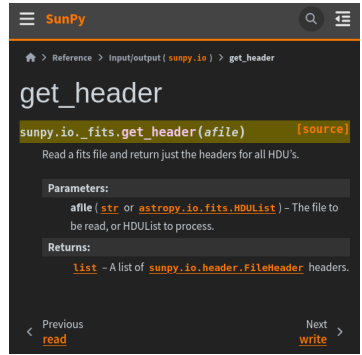
- Touches **reproducibility**, **collaboration**, **clarity**, **transferability**, and **transparency**

Let's look at some forms of documentation

Best Practice: Documentation (Cont.)

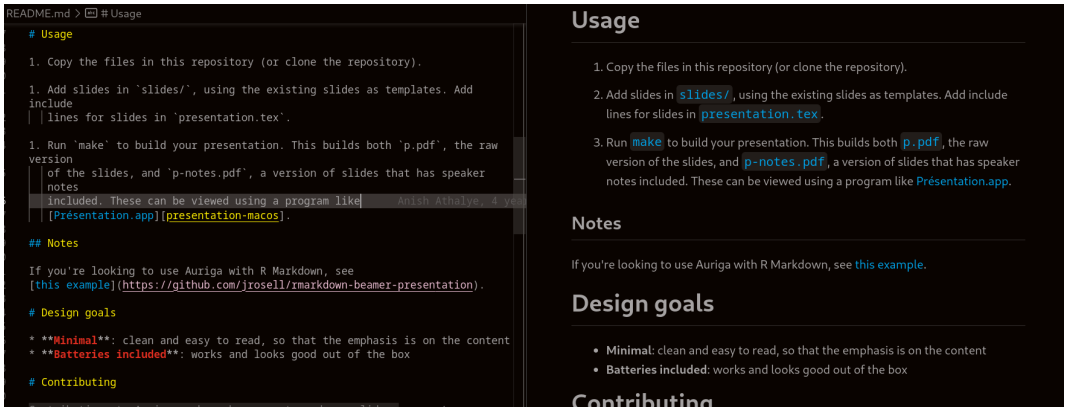
You can write documentation directly in the source code itself or at a higher level outside the source itself.

```
def get_header(afile):
  """
  Read a fits file and return just the headers for all HDU's.

  Parameters
  ----------
  afile : `str` or `astropy.io.fits.HDUList`
    The file to be read, or HDUList to process.

  Returns
  -------
  `list`
    A list of `sunpy.io.header.FileHeader` headers.
  """
  if isinstance(afile, fits.HDUList):
    hdulist = afile
    close = False
  else:
```



Best Practice: Documentation (Cont.)

You can write documentation directly in the source code itself or at a higher level outside the source itself.



Conclusions: Other Concepts

- Testing
- Issue Tracking
- Automation (CI/CD)
- Peer-Programming
- and more

These can further help with **reproducibility** and other goals mentioned

Conclusion: Resources

- The US Research Software Engineer Association (US-RSE)
- Better Scientific Software
- NSF-funded Intersect Training (RSE Training)

US-RSE

Better Scientific Software

Intersect RSE Training

Any Questions?

exoticdft | akoufos@sun.stanford.edu

Join the US-RSE

USRSE'24 - Albuquerque, NM

Poster Version