

Writing and Sharing Computational Analyses in Jupyter Notebooks

Presented by: Peter W. Rose
Structural Bioinformatics Lab
San Diego Supercomputer Center, UC San Diego
pwrose@ucsd.edu



Overview

- Motivation: Open & Reproducible Research
- Ten Simple Rules
- Demo on CyVerse/VICE
- Tools to Support Reproducible Workflows

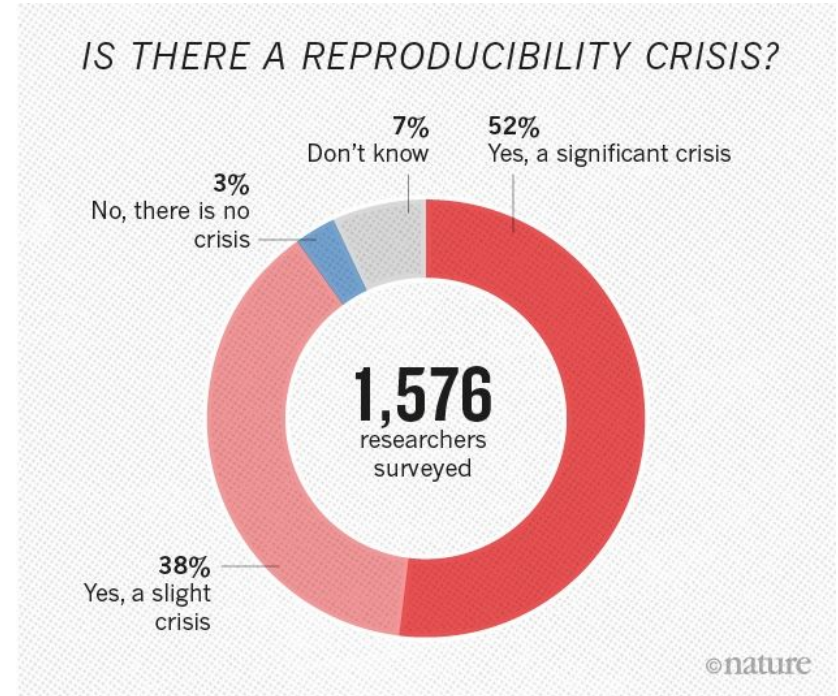
Reproducibility Crisis?

“More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments.”

Nature, 2016, M. Baker, 1,500 scientists lift the lid on reproducibility

“Nature journal editors ... will, on a case-by-case basis, ask reviewers to check how well the code works.”

Nature, 2018, Does your code stand up to scrutiny?



Reproducibility*

obtaining consistent results using

the same input data,

computational steps, methods, and code,

and conditions of analysis

Reusability

obtaining new results using

different input data or parameters,

the same computational steps, methods, and code,

and conditions of analysis

Replicability*

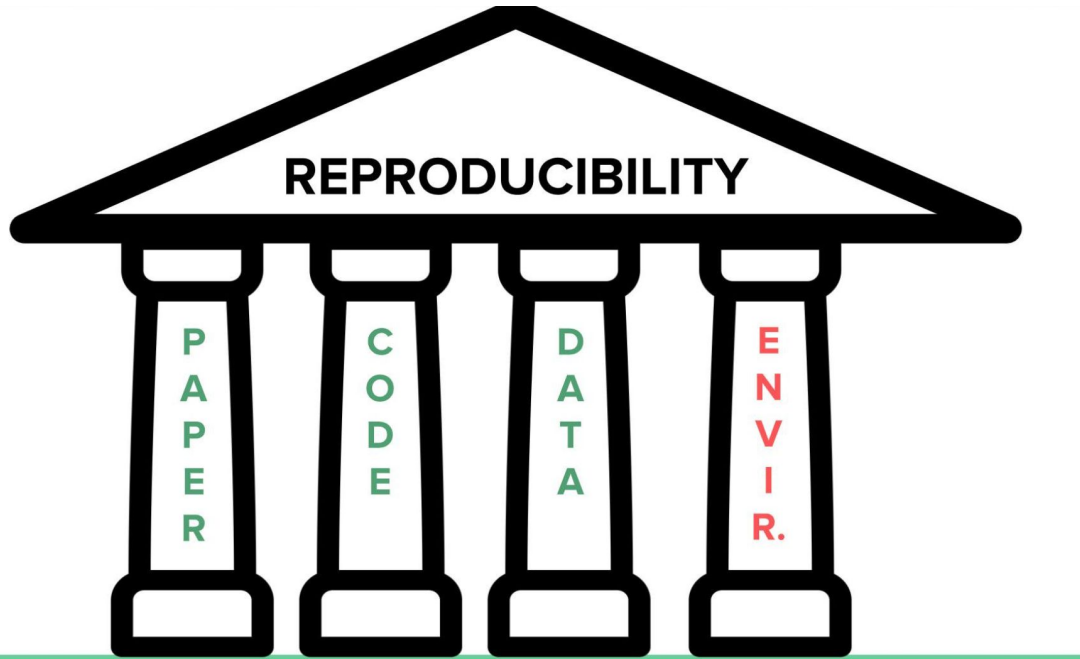
obtaining consistent results across

studies aimed at answering the

same scientific question,

each of which has obtained its own data

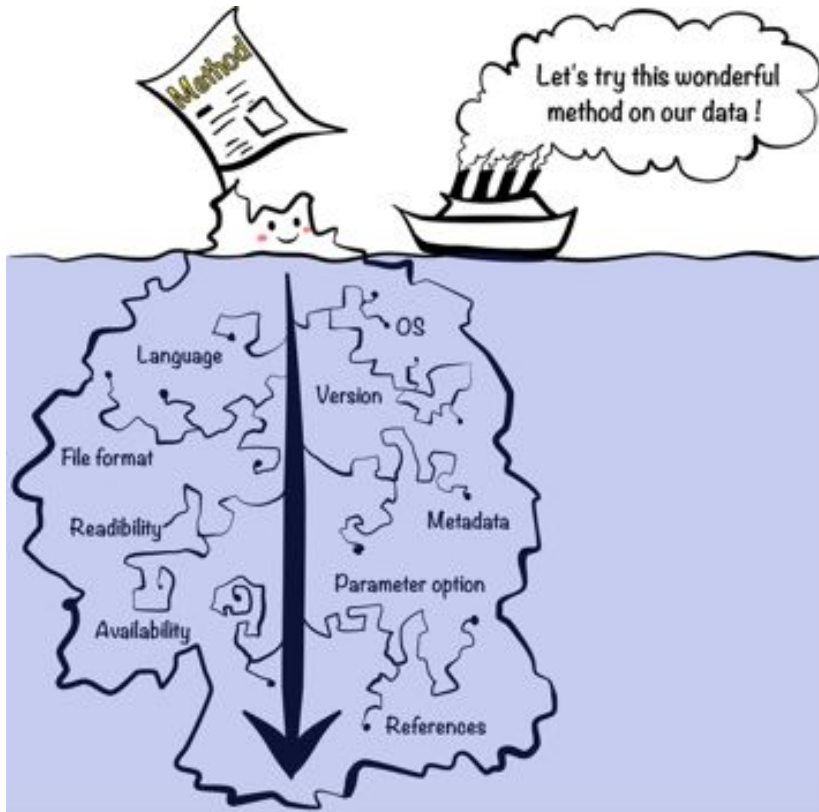
Four Pillars of Reproducible Research



Open Science

- Open access publications
- Open data
- Open source code
- Open execution environment

Barriers to Reproducibility and Reusability

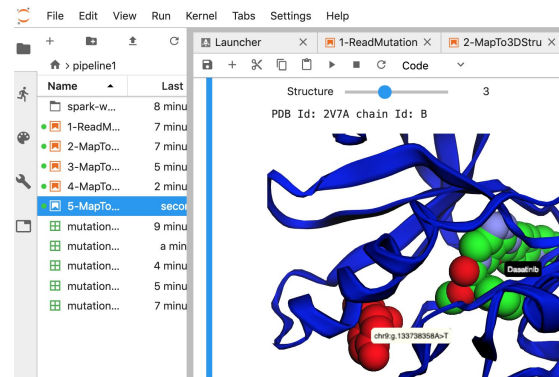
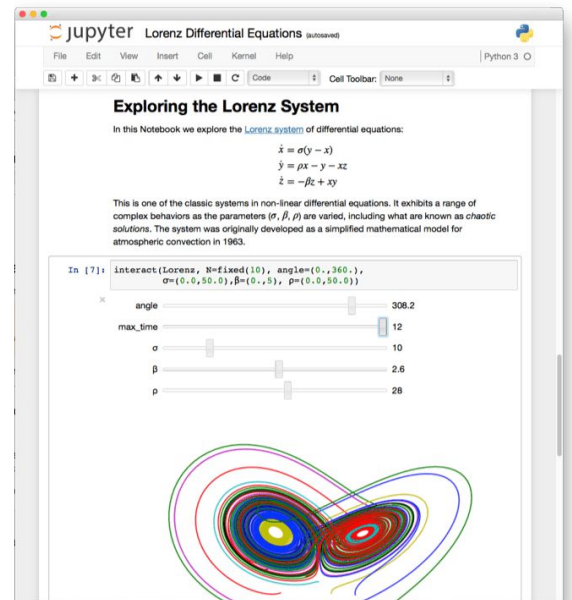


- Missing or incomplete documentation
- Distribution is missing files
- Missing third party package
- Dependencies failed to build
- Runtime error
- Internal compiler error
- My last week:
 - samtools: error while loading shared libraries: libbz2.so.1.0: cannot open shared object file
 - error while loading shared libraries: libz.so.1: failed to map segment from shared object: Operation not permitted
 - /lib64/libc.so.6: version `GLIBC_2.14' not found

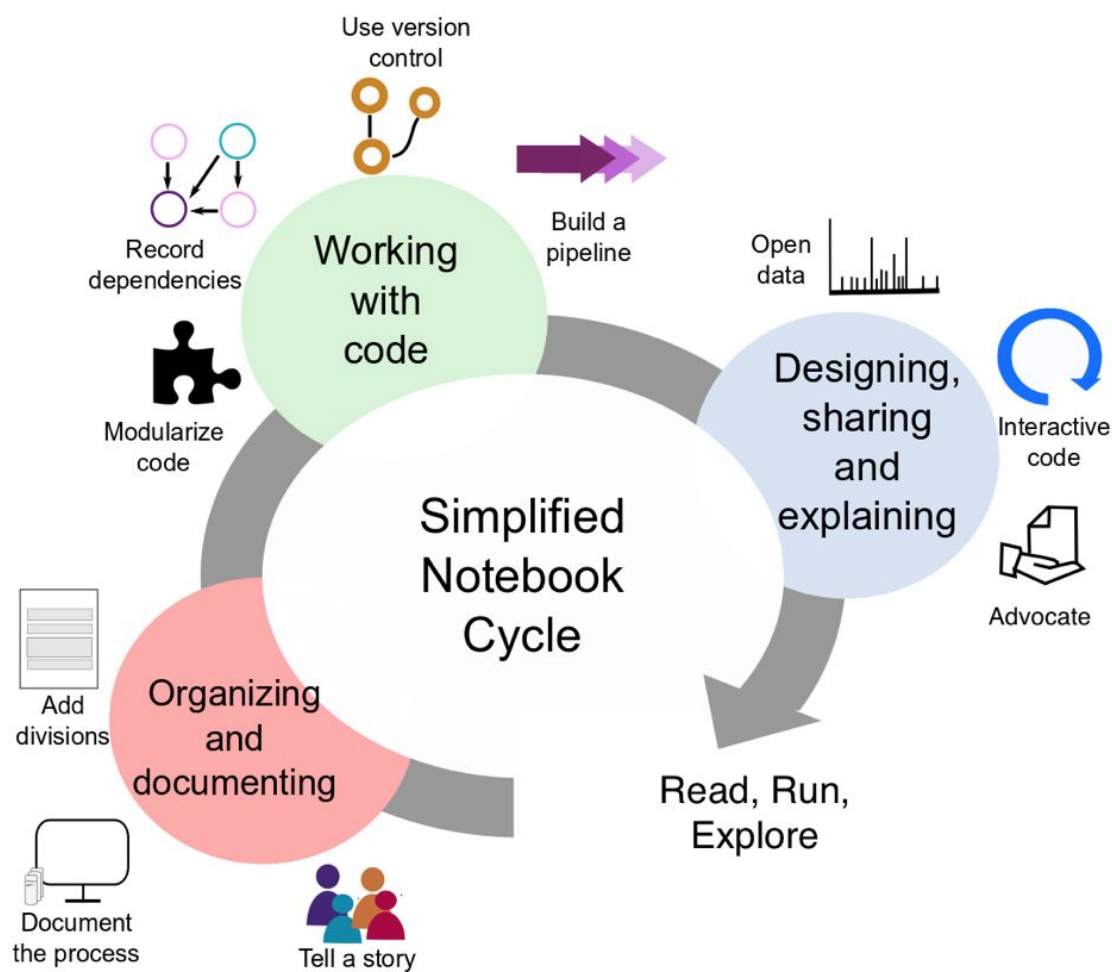
Jupyter Notebook



- Important tool in support of reproducible workflows
- Combines live code, equations, images, multimedia, visualizations, and narrative text
- Supports over 40 programming languages
- ~5.5 million Jupyter Notebooks on GitHub in Oct. 2019 (<https://github.com/parente/nbestimate>)
- Jupyter Lab, next generation user interface

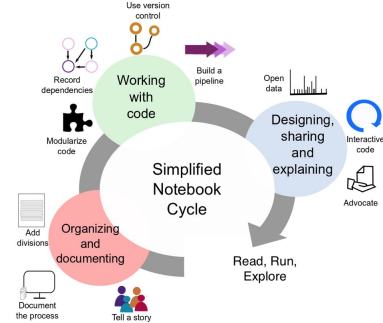


Ten Simple Rules



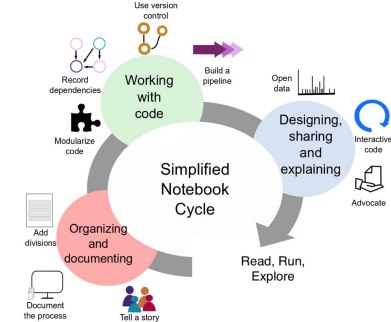
Organizing and Documenting

- Rule 1: Tell a Story for an Audience
 - Beginning - introduce topic
 - Middle - describe steps
 - End - interprets results
 - Describe not just what you did, by why you did it, how the steps are connected, and what it all means.
 - Adjust your description depending on the intended audience
- Rule 2: Document the process, not just the results
 - Add descriptive notes, e.g., why a particular parameter was chosen
- Rule 3: Use cell divisions to make steps clear
 - Avoid long cells
 - Limit each cell to one meaningful step
 - Split long notebooks into a series of notebooks
 - Keep a top-level index notebook with links to the individual notebooks



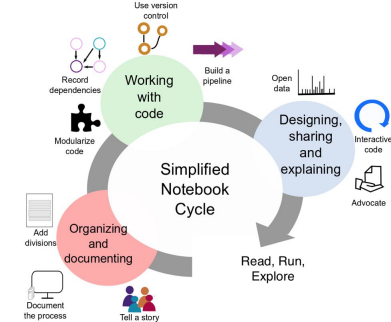
Working with Code

- Rule 4: Modularize Code
 - Use functions instead of duplicating code cells
- Rule 5: Record Dependencies
 - Manage your dependencies explicitly from the start using a tool such as
 - Conda's `environment.yml`
 - pip's `requirements.txt`
- Rule 6: Use Version Control
 - Consider using a public repository from the beginning of a project
 - Tie research results to specific software versions
- Rule 7: Build a Pipeline
 - Design notebooks with reuse in mind (different input data and parameters)
 - Define key input data and parameters at the top of each notebook
 - Break long notebooks into smaller notebooks that focus on one or a few analysis steps.

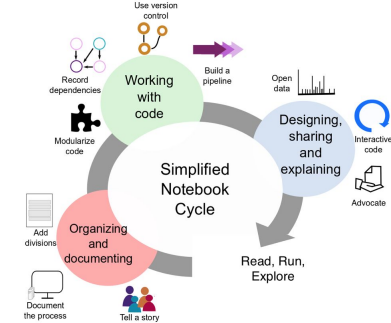



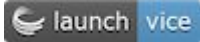
Sharing, explaining

- Rule 8: Share and Explain Your Data
 - Share your data in a repository with a persistent identifier, e.g., DOI or ARK
 - Bio repositories, e.g., NCBI, Ensemble, PDB
 - General repositories, e.g., Zenodo <https://zenodo.org/>
 - Small datasets can be stored in GitHub with your source code (< 50MB)
 - E.g., in a /data folder
 - Very large datasets
 - Consider using a sample of the data and a link to the original data
 - Save intermediate data after data processing
 - E.g., in /intermediate_data folder
 - Can be used to verify each step in a workflow



Sharing, explaining cont.



- Rule 9: Design your notebooks to be read, run, and explored
 - Git repository
 - Add a descriptive README file
 - Add a LICENCE file (liberal licence, e.g., MIT, Apache 2, BSD)
 - Add a static HTML/PDF file of your notebooks for long-term preservation
 - Add a badge/link to launch notebooks in cloud environment (no installation required)
 - Binder (<https://mybinder.org/>) 
 - CyVerse/VICE (<https://learning.cyverse.org/projects/vice/en/latest/>) 
 - Consider using ipywidgets to add menus or sliders to enable interactive exploration of parameters

Sharing, explaining cont.

- Rule 10: Advocate for open research
 - Apply what you learned in this tutorial in your own research and be an advocate for open and reproducible research in your lab or workplace
 - Publish a fully reproducible paper! Create all figures, data tables, and all other computational results using Jupyter Notebook and deposit in GitHub.



Brad Voytek  @bradleyvoytek · 20 Apr 2018

Our lab's moving to this model: publish "static PDF" papers as expected, but also a shadow, interactive [@ProjectJupyter](#) version alongside that has all code to process, analyze, and visualize data.

"The Scientific Paper Is Obsolete" featuring [@fperez_org](#)



The Scientific Paper Is Obsolete

Here's what's next.

[theatlantic.com](#)

Demo: Running Jupyter Lab on VICE

Visual & Interactive Computing Discovery Environment

Ten Simple Rules Example Notebooks:

<https://github.com/jupyter-guide/ten-rules-jupyter>



Tools and Infrastructure



Computational Notebooks:
Jupyter Notebook
Jupyter Lab



git

Version-control system
for tracking changes in
source code



GitHub

Source code
repository



Open-source package
and environment
management system



docker

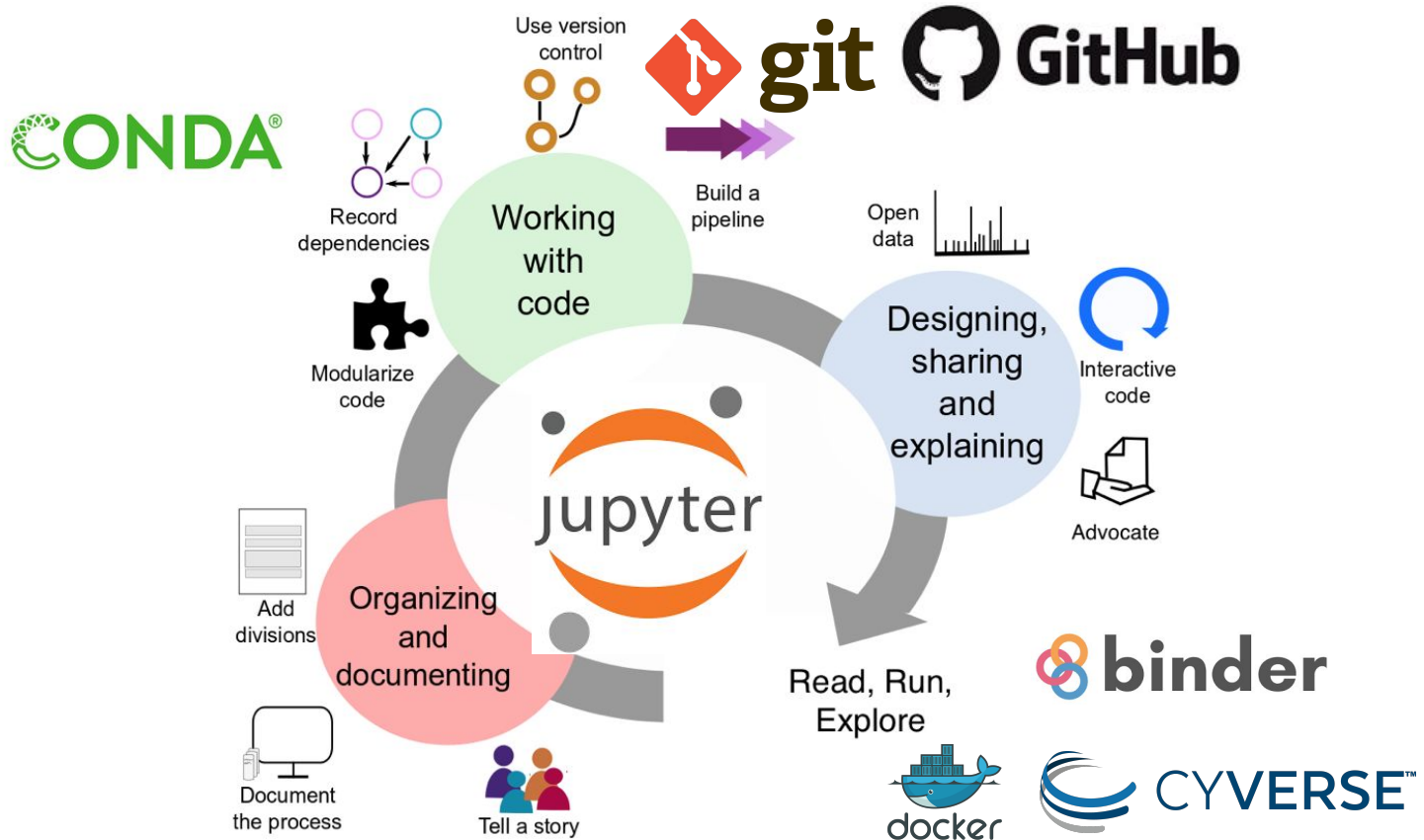
Docker Image
encapsulates OS with
executable code

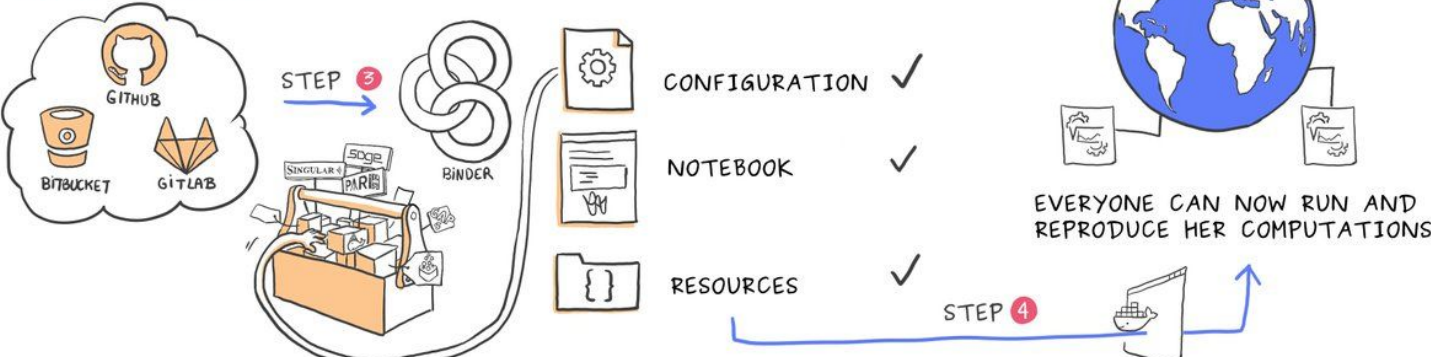
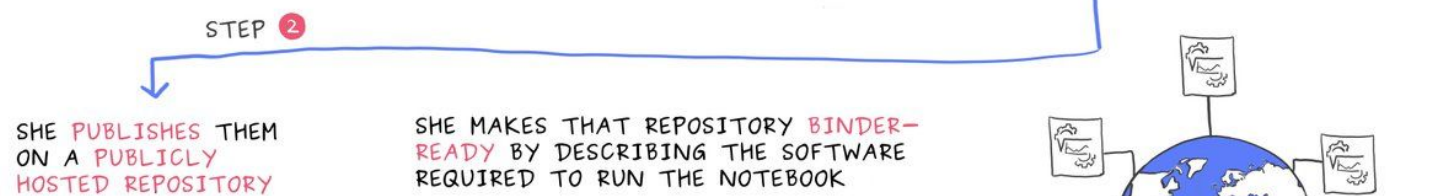


CYVERSE™

Cloud environment to run
computational notebooks
(including RStudio)

Tools for Reproducible Workflows





Learn More

- CyVerse Learning Institute: Webinars & Workshops
<https://www.cyverse.org/learning>
- How to setup a CyVerse/VICE application
<https://learning.cyverse.org/projects/vice/en/latest/>
- Ten Simple Rules Example Notebooks
<https://github.com/jupyter-guide/ten-rules-jupyter>
- Guide for Reproducible Research and Data Science in Jupyter Notebooks
<https://github.com/jupyter-guide/jupyter-guide>
- A Practical Introduction to Reproducible Computational Workflows (GitHub, Conda, Binder)
<https://github.com/ISMB-ECCB-2019-Tutorial-AM4/reproducible-computational-workflows>

Take the Challenge!


Publish a fully reproducible paper!

Create all figures, data tables, and all other computational results using Jupyter Notebook and deposit in GitHub.

Deploy Notebooks on CyVerse/VICE & Binder

Acknowledgements

Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks

Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H. Nguyen, Sara Brin Rosenthal, Fernando Pérez, Peter W. Rose 

Published: July 25, 2019 • <https://doi.org/10.1371/journal.pcbi.1007007>

Funding for the “Reproducible Research and Interactive Education - Application of Jupyter Notebooks” workshop was provided by the San Diego Supercomputer Center at UC San Diego.

Thanks to the CyVerse Team!



CyVerse is supported by the National Science Foundation under Grants No. DBI-0735191, DBI-1265383, and DBI-1743442.



Cold
Spring
Harbor
Laboratory

