A Practical Introduction to Reproducible Computational Workflows

Make your code reproducible by anyone, anywhere

Tools and Infrastructure

Computational Notebooks:
Jupyter Notebook
Jupyter Lab

Cloud environment to run computational notebooks (including RStudio)

Source package and environment management system

Source code repository

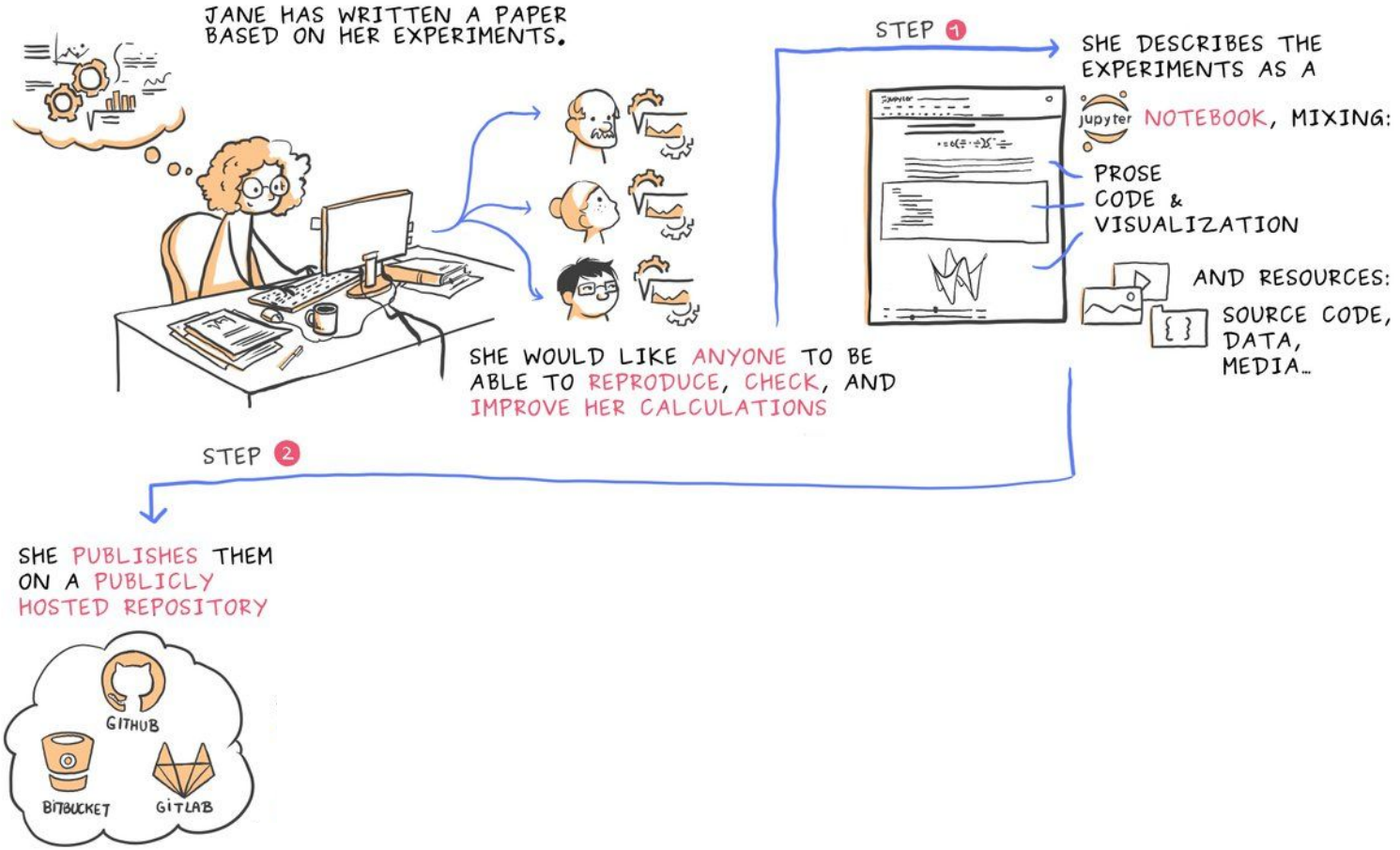
The binder Project

A community that builds **free and open-source** tools
for **reproducible, sharable scientific environments**
that are **workflow- and platform-agnostic**.

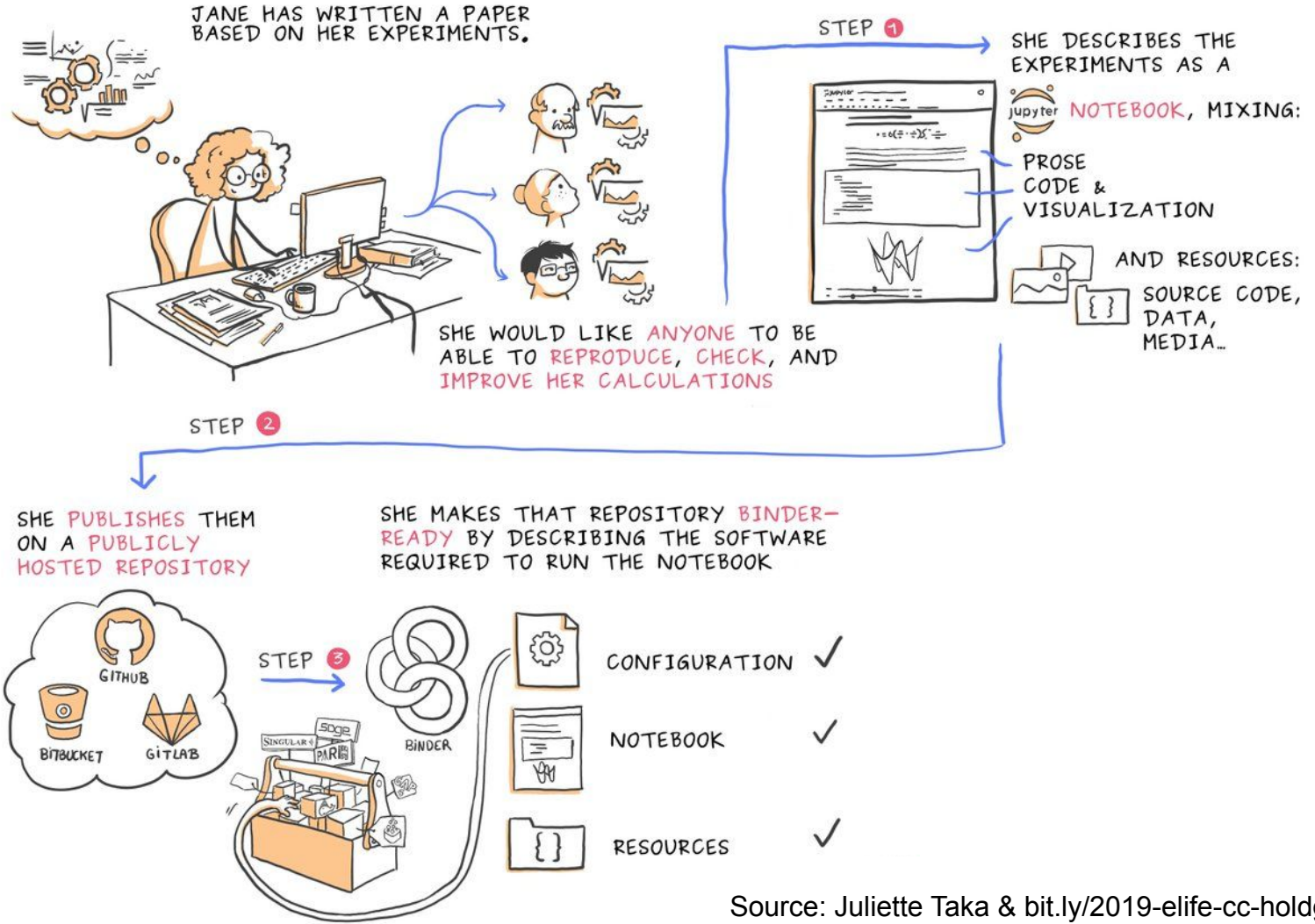
Source: Chris Holdgraf, bit.ly/2019-elife-cc-holdgraf

Source: Juliette Taka & bit.ly/2019-elife-cc-holdgraf

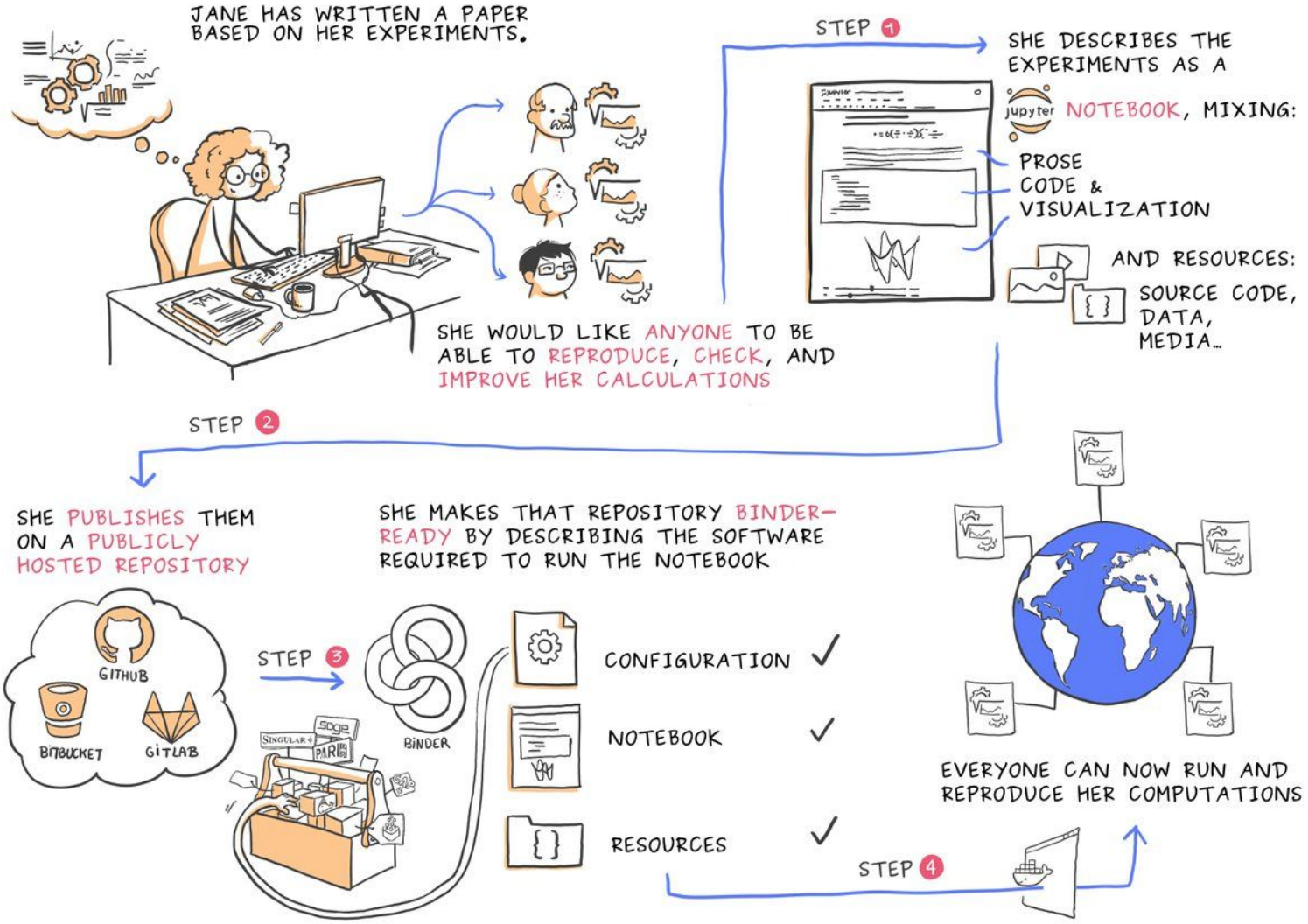
Source: Juliette Taka & bit.ly/2019-elife-cc-holdgraf



Source: Juliette Taka & bit.ly/2019-elife-cc-holdgraf



Source: Juliette Taka & bit.ly/2019-elife-cc-holdgraf

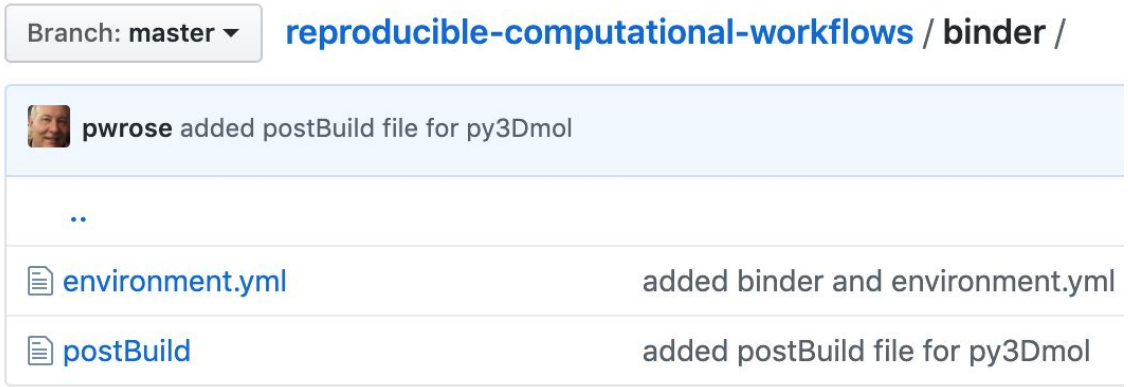

JANE HAS WRITTEN A PAPER BASED ON HER EXPERIMENTS.
STEP 1
SHE DESCRIBES THE EXPERIMENTS AS A
jupyter NOTEBOOK, MIXING:
PROSE
CODE &
VISUALIZATION
AND RESOURCES:
SOURCE CODE, DATA, MEDIA…
SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS
STEP 2
SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY
SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK
GITHUB
BITBUCKET
GITLAB
STEP 3
BINDER
CONFIGURATION ✓
NOTEBOOK ✓
RESOURCES ✓
EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS
STEP 4

How to configure a Repository for Binder?

“Binderize a repository”

Mybinder.org uses the files in **/binder** folder to create a conda environment.

The software dependencies are specified in environment.yml.



Documentation: https://mybinder.readthedocs.io

environment.yml

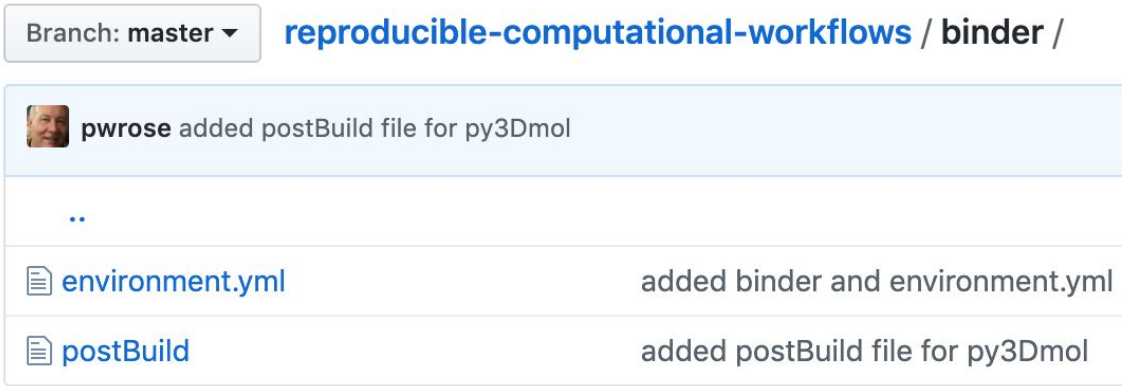
```yaml
name: ismb2019

channels:
  - conda-forge

dependencies:
  - python=3.7
  - numpy
  - pandas
  - matplotlib
  - seaborn
  - scikit-learn
```

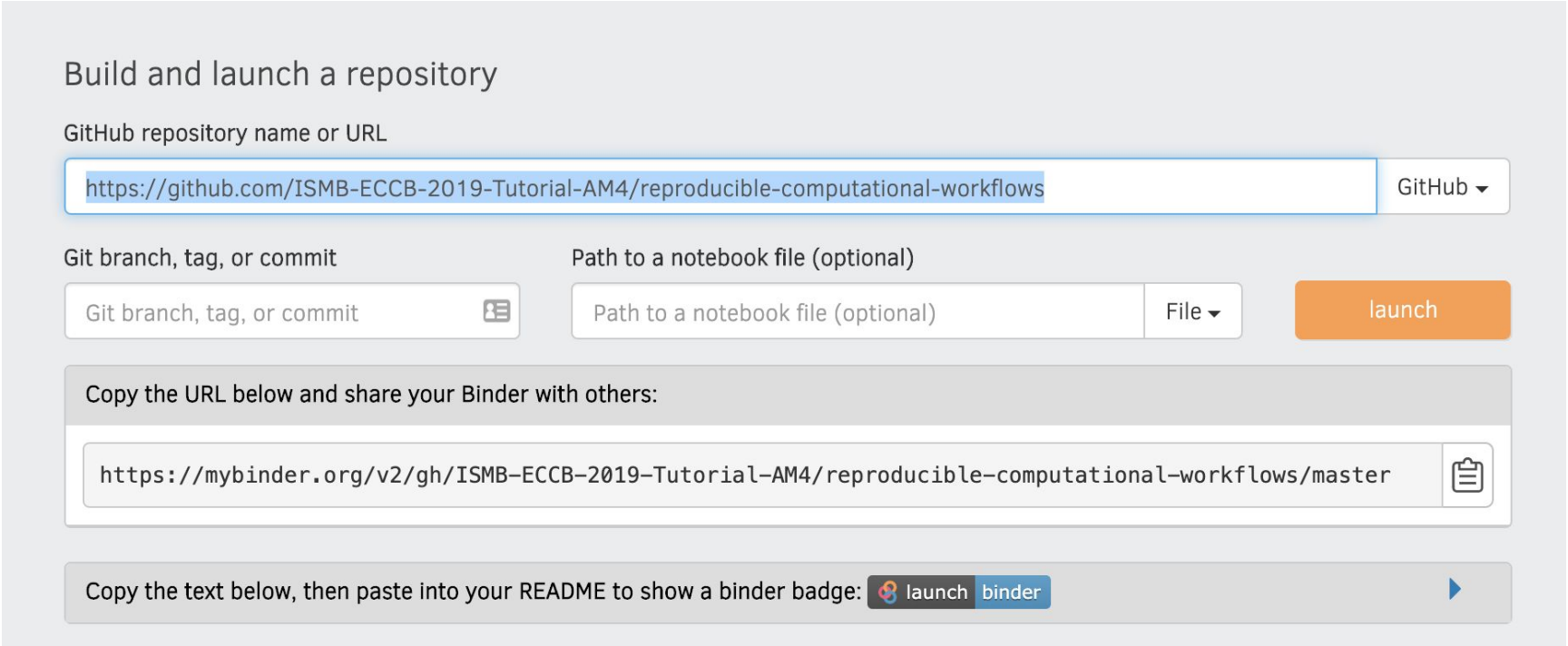
Jupyter Lab Plugins

- Some interactive packages require a Jupyter Lab plugin, e.g.
  - py3Dmol
  - ipywidgets
  - These plugins are specified in the postBuild file
  - https://github.com/ISMB-ECCB-2019-Tutorial-AM4/reproducible-computational-workflows/binder/postBuild



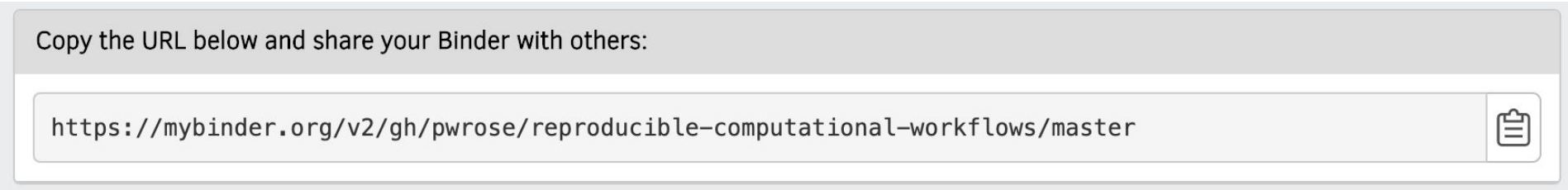
How to Launch Repo on https://mybinder.org?

https://github.com/ISMB-ECCB-2019-Tutorial-AM4/reproducible-computational-workflows

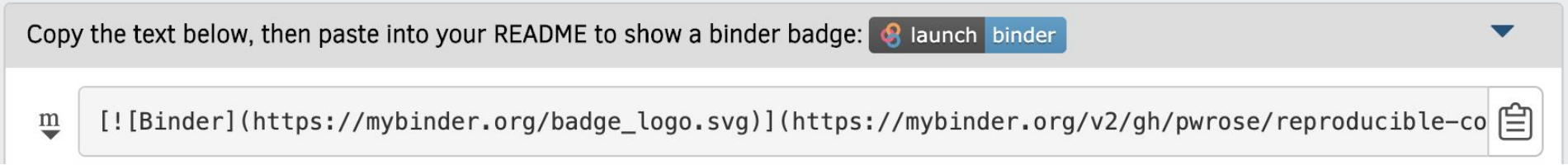


Make your code reproducible by anyone, anywhere

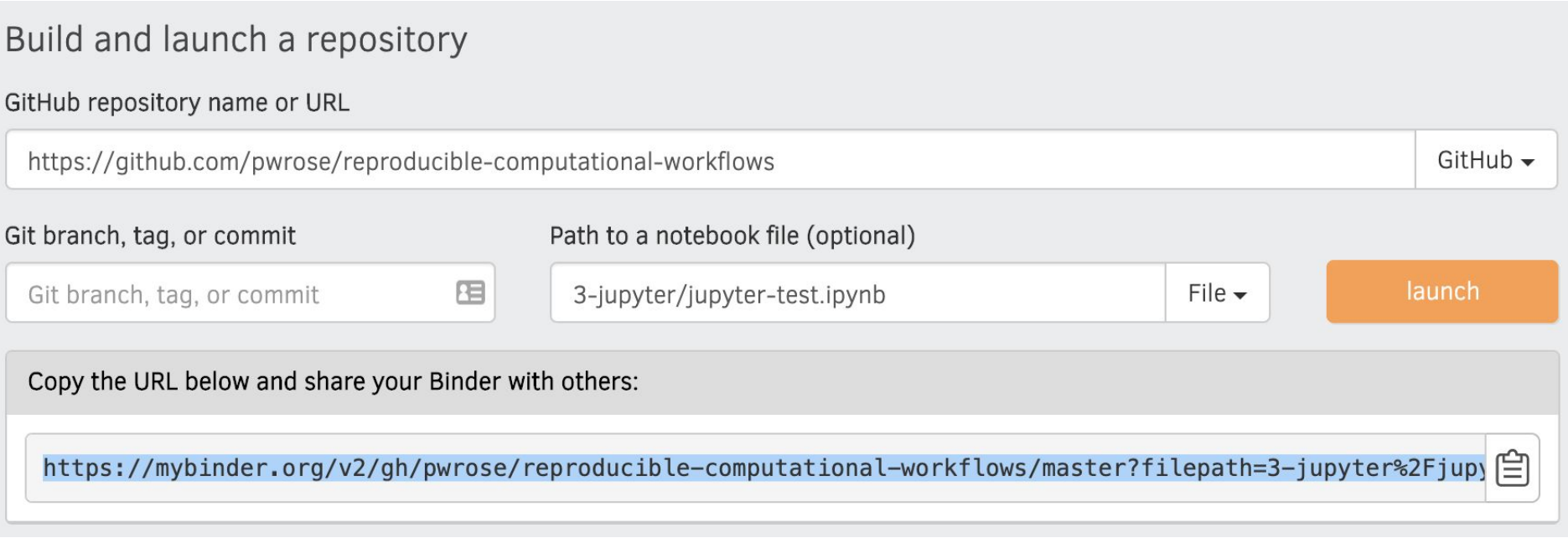
Share a link with your collaborators



Add a Jupyter Notebook launch button to your repository README.md file



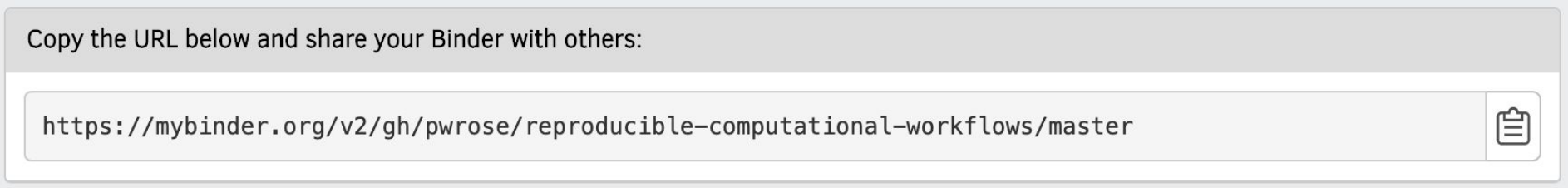
Launch Link for a Specific Notebook



https://mybinder.org/v2/gh/pwrose/reproducible-computational-workflows/master?filepath=3-jupyter%2Fjupyter-test.ipynb

How to create a Launch Link for Jupyter Lab?

By default binder links to Jupyter Notebook



To launch Jupyter Lab append **?urlpath=lab** to the launch link:
https://mybinder.org/v2/gh/pwrose/reproducible-computational-workflows/master?urlpath=lab

To launch a specific notebook in Jupyter Lab append **?urlpath=lab/tree/path-to-notebook/notebook.ipynb** to the launch link:

https://mybinder.org/v2/gh/pwrose/reproducible-computational-workflows/master?urlpath=lab/tree=3-jupyter/Jupyter-test.ipynb

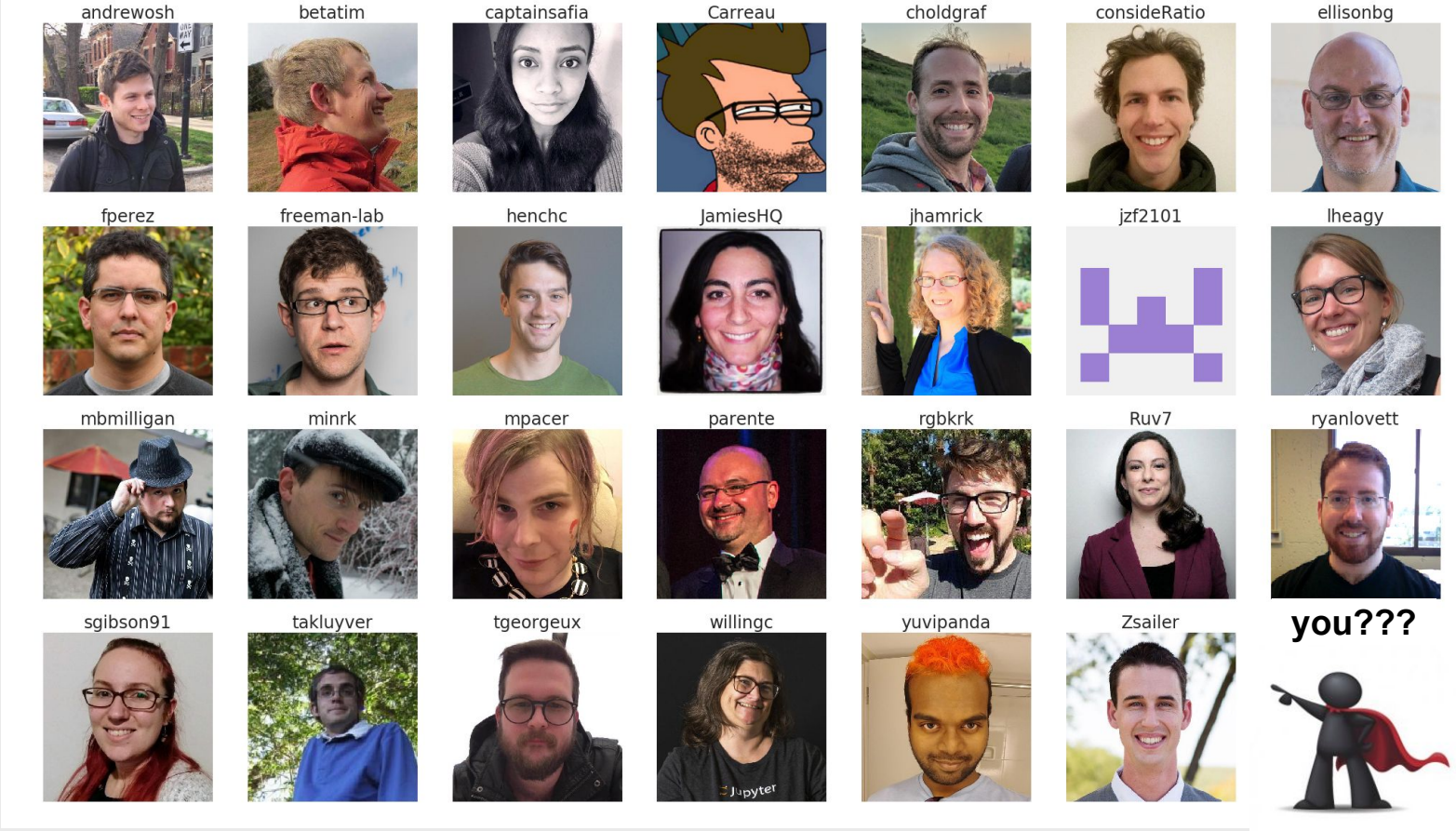
Hands-on Session
Binderize your Repository

https://github.com/ISMB-ECCB-2019-Tutorial-AM4/reproducible-computational-workflows/5-binder/README.md

Other Options to Share Jupyter Notebooks

- CyVerse VICE (Visual Interactive Computing Environment)
  - http://learning.cyverse.org/projects/vice/en/latest/index.html
  - **Free**, account sign-up required
  - More CPUs, memory and persistent storage (default 100GB)
  - Run notebooks for up to 48 hours
  - Suitable for production and compute intensive tasks
  - Supports Jupyter Lab, RStudio, Shiny applications
  - Funded by the US National Science Foundation
- Demo
  - https://github.com/sbl-sdsc/mmtf-genomics

Other Options to Share Jupyter Notebooks

- Commercial
  - Suited for ad hoc sharing of single Notebooks, not reproducible research
  - Google Collaboratory (uses Google Drive to share single Notebooks)
    - https://colab.research.google.com
    - Pros: more CPU, RAM and GPUs
    - Cons: limited options to customise execution environment
  - Microsoft Azure Notebooks
    - https://notebooks.azure.com/

A round of Applause for the Jupyter/Binder Team