The Open road: A case study of reproducible research in plant pathology

Zhian N. Kamvar and Sydney E. Everhart — Department of Plant Pathology, University of Nebraska, Lincoln, NE, USA

Closed Research

Traditional publishing requires only the final manuscript to be published. The data and analyses are not shared with the manuscript. Data and analyses cannot be easily inspected or assessed for accuracy.

Open Research

Unlike closed research, open encourages sharing of data, and analyses along with the publication. shared data and analyses can provide a wealth of services to the scientific community including a wealth of materials for improved and up-to-date training, readily available data sets for meta-analyses, and increased potential for collaboration.
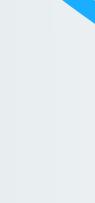
Reproducible Research

One of the benefits of open research is that it is simple to show that it is also reproducible. If anyone can take your data, run your analyses/code, and obtain the same results, then the research is reproducible!

Open and reproducible practices can help increase trust in plant pathologist decisions that directly influence management decisions by mitigating anomalous results due to human error.

Open Data

- Share with an open license (e.g. CC0)
- Archive in a repository with a DOI
- Use in future analyses

License information shows who owns the work and how it can be used.

zenodo — https://zenodo.org
osf.io — https://osf.io
DRYAD — https://datadryad.org
figshare — https://figshare.com

Archives such as Zenodo, Open Science Framework, Dryad, and Figshare will save permanent digital copies of your data and code.

Data stored in a plain text format such as csv is best

Data

Open Source Code

- Use version control — a lab notebook for code
- Archive in repository with DOI
- Allow reviewers to vet your work

zenodo — https://zenodo.org
osf.io — https://osf.io
DRYAD — https://datadryad.org
figshare — https://figshare.com

Live repositories such as GitHub can synch with archive repositories to host both live and archived versions of code

Analysis/Code

Open Access Pre-Print Publication

- Solicit feedback from peers
- Available to anyone in the world

PeerJ — https://peerj.com/preprints
bioRxiv — https://www.biorxiv.org/

This unformatted version can continue to be updated after peer review
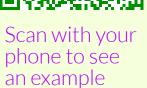
V1
V2

Manuscript/Revisions

Open Tools

The data and code are valuable for assessing the process of analysis, but does not ensure that it can be re-run by describing or providing the details of the software used for analysis, researchers can demonstrate computational reproducibility with a low barrier for entry. This can take several forms:

Research Compendium

This is a text file that contains a description of your operating system, and all the software packages used. For example, in R, this is accomplished with the command `sessionInfo()`
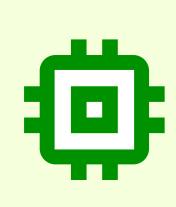
Scan with your phone to see an example

Installation Script

This file contains all the commands necessary to install the software packages for analysis on a brand new computer, usually named `install.R` or `install.py`

Virtual Machine

A virtual machine is a complete operating system encapsulated within a disk image. The software called Docker makes these portable and configurable with a script called a **Dockerfile**. This option provides complete stability to ensure reproducibility

System-Independent Analysis

Code that works on your computer may not work anywhere else. Use these strategies to prevent this situation.

- **Run and test analysis on a colleague's computer**
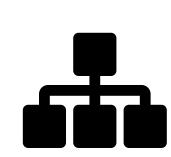- **Run and test analysis on the cloud**
- binder — https://mybinder.org — Interactive sessions
- https://docker.com — Docker container with continuous integration for automated analysis
- https://circleci.com

Tips and Tricks

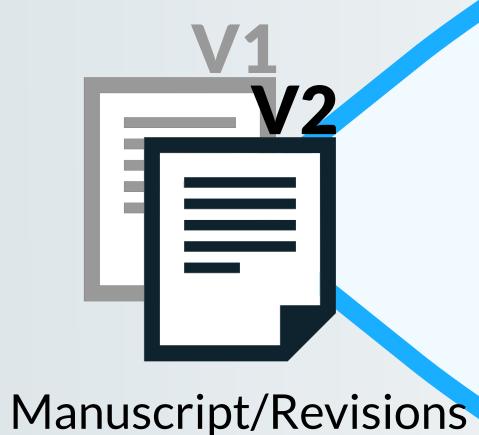Start small! Begin by sharing your data and code with your paper and work up from there.

DO cultivate a workflow. Use automated reports like RMarkdown.
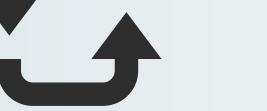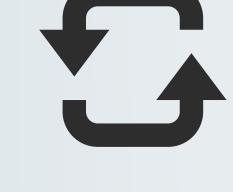
DO use others' workflows and templates for inspiration.

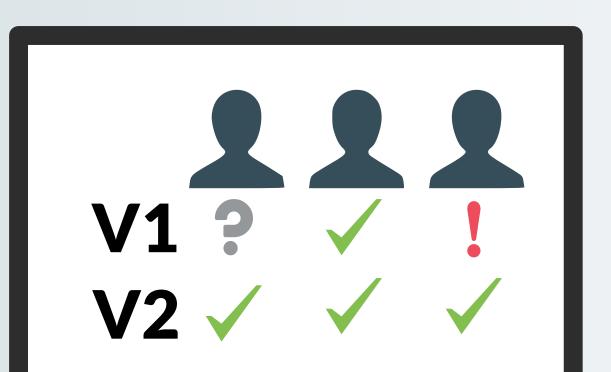DO NOT worry too much about your code style. Everyone has a different approach.

Less Automated — **WORKFLOW MANAGEMENT** — Fully Automated

V1 / V2

V1 ? ✓ !
V2 ✓ ✓ ✓

Peer Review

Official Publication

Case Study: *Sclerotinia sclerotiorum*

https://github.com/everhartlab/sclerotinia-366

We assessed population genetic structure for 366 samples of *S. sclerotiorum*. The manuscript and all analyses were written in RMarkdown. These were stored on GitHub, and run on circle-ci with a dedicated docker container and archived on the Open Science Framework.

The Dockerfile is used to rebuild the analysis anywhere; including the cloud and your computer.

https://osf.io/ejb5y — Results are saved and archived at the OSFramework
Permanent link added to manuscript — Analysis is re-run after any change to the source code
https://peerj.com/articles/4152/ — Raw Data + RMarkdown + Dockerfile
Manuscript sent for peer-review — Manuscript regenerated with figures and tables.