Reproducible Computational Research with Popper Workflows

Anders Poirel, Ivo Jimenez

CROSS 2020 Research Symposium

Baskin Engineering UC SANTA CRUZ | UC SANTA CRUZ | CROSS | CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE

Research in computational fields is too often difficult to replicate, even when code and data are made available. Why?

Obstacles to Reproducibility

- Projects rely on **complex software dependencies** to use statistical or machine learning tools
  - Dependencies break when running the code on another computer
- Data processing involves executing **multiple dependent steps**
  - If not documented, running the code requires guesswork

How Does Popper Help?

- **Containerization** helps manages dependencies while eliminating environment differences
- A **workflow definition language** documents explicitly steps in a workflow so that it can be replicated with a single command

However…

- Popper requires knowledge of many tools (Docker, scripting...)
- This is a steep learning curve for many potential users in computational fields

Popper

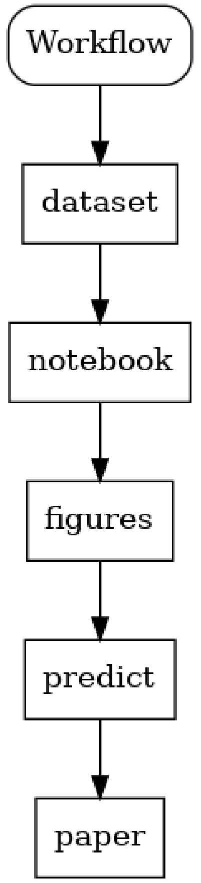
These users would benefit from an easy starting point for adopting the “Popper” approach

Tools to help Python and R users adopt Popper

Workflow Guides

A detailed how-to for common tasks in computational research with Popper

- Managing **software dependencies**
- Running a **computational notebook**
- Using a **easy to** project structure



Cookiecutter Templates

Using the `cookiecutter` utility, templates to **bootstrap** a Popper workflow with sensible defaults and Docker images designed to work well with Popper

Workflow Examples

Sample machine learning workflows in Python and R showcasing **common tasks** and **best practices**

Anders Poirel – apoirel@ucsc.edu | Mentor: Ivo Jimenez – ivotron@ucsc.edu | Find this project at popper.readthedocs.io !