Tools: Versioning

Dr. David Koop
Tools

• We have seen specific tools that address particular topics:
  - Versioning and Sharing: Git, Github
  - Data Availability and Citation: DOIs, Dryad, DataONE, figshare
  - Virtual Machines and the Cloud: Xen, Parallels, EC2
  - Containers: Docker
  - Scientific Workflows: Pegasus, Kepler, VisTrails, Taverna
  - Provenance: PDIFF, Analogies, VisComplete

• More at ReproMatch
Packaging

• If I want to replicate/reproduce something,
  - How do I ensure that I have everything I need?
  - How do I package it up so that someone else can access it?

• Two Tools
  - CDE
  - ReproZip
CDE: Distributing Code

Amount of pain your users suffer

Amount of pain you (the author) must suffer

∞

[Guo and Engler, 2011]
Creating a package with cde

Timeline

[Guo and Engler, 2011]
Executing a package with cde-exec

Timeline

executing a package with cde-exec

D. Koop, CIS 602-01, Fall 2016
Creating a package with `cde`

```bash
 cd /home/pg/expt/
cde python predict_weather.py
```

[Guo and Engler, 2011]
CDE Limitations

- Packages might be incomplete
- Execution is slower (2% - 30%)
- Cannot emulate custom hardware
- Only x86 → x86, Linux 2.6 → Linux 2.6
Pillars of Reproducibility

[Reproduducibility]

[Rampin et al., 2015]
ReproZip: Include all pillars

necessary data files, libraries, environment variables, etc. required to reproduce your data analysis

open, unpack, and reproduce anywhere, anytime!

[Rampin et al., 2015]
ReproZip Packing and Unpacking

Packing Step (Linux)

reprozip trace

reprozip pack

Unpacking Step (Linux, Mac OS X, Windows)

reprounzip setup

reprounzip upload

reprounzip run

reprounzip download

reprounzip destroy

[Rampin et al., 2015]
ReproZip Packing

Computational Environment $E$ (Linux)

Data Analysis; Software; Environment

Executing

reprozip

Tracing
ptrace + SQLite

Configuring

Configuration File

Creating Configuration

Data Analysis Provenance

Data
- Input files, output files, parameters

Workflow
- Executable programs and steps

Environment
- Environment variables, dependencies, software packages, ...

Rampin et al., 2015
ReproZip Unpacking

Computational Environment $E'$ (Linux, Windows, OS X)

- Unpacks and reproduces from a single directory
  - Linux
- Unpacks in a single directory and builds a full system environment
  - Linux
- Unpacks in a virtual machine using Vagrant
  - Linux | Mac OS X | Windows
- Unpacks in a Docker container
  - Linux | Mac OS X | Windows

[Rampin et al., 2015]
ReproZip advantages over existing tools

• Focus on **without-forethought** reproducibility
• Other tools: PTU, CARE, CDE
• Portability: runs using VMs or containers
• Extensibility: may port to other environments using other unpackers
• Modifiability: identifies inputs, outputs, parameters
• Usability: command-line tool and control over the trace

[Rampin et al., 2015]
ReproZip Limitations

• Only packs experiments in Linux
• Only detects information about software packages in Debian and Fedora-based environments
• Does not allow reproducibility of non-deterministic processes
• Does not save state

[Rampin et al., 2015]
Assignment 2

• http://www.cis.umassd.edu/~dkoop/cis602/assignment2.html
• Keep your project on Github, keep images on Docker Hub
• Put a link to your Docker Hub images in your Github README.md
• Questions?
Projects

• Survey:
  - Everyone doing this option should have received an email
  - Three of you did not submit a preference according to my email and were assigned papers
  - Image/figure checks:
    • Eyeball method
    • Something more precise: http://www.pyimagesearch.com/2014/09/15/python-compare-two-images/

• Research:
  - Similar questions about checking data
What happens if code or data changes?
Automated Capture of Experiment Context for Easier Reproducibility in Computational Research

A. Davison
Steps in capturing experiments in Sumatra

1. Create a new record.
2. Has the code changed?
   - Yes: Raise exception.
   - No: Find dependencies.
3. Error: Store the code change policy.
   - Diff: Run simulation/analysis.
   - Find new files.
   - Add tags.
   - Record time taken.
   - Find dependencies.
   - Get platform information.
   - Run simulation/analysis.
   - Record time taken.
   - Find new files.
   - Add tags.
   - Save record.

D. Koop, CIS 602-01, Fall 2016
Advantages?
Advantages

• Separates out information about configuration/parameters
• Detects changes in code and data
• Web-based interface to examine information and versions
• Set a general run command
• Annotations
Disadvantages?
Disadvantages

• Command-line based
• Focused more on personal reproducibility…
• Storage?
Linking Tools and Provenance

- Provenance often tracks different versions, different files
- Can we introspect programming languages?
noWorkflow: Capturing and Analyzing Provenance of Scripts

J. F. Pimentel, L. Murta, V. Braganholo, D. Koop, J. Freire