Best Practices for Reproducible Research

Darin London, Office of Research Informatics
Research Application Development

- Share the data
- Share the code
- Share the compute

Share your Data with the World

- First roadblock to reproducibility, lack of access to data
- Journals are beginning to require data be made available
- Consider keeping a journal of data provenance (where you got it, when you got it, its md5/sha1 hashsums, what processes were used to produce it, …), and storing it with the data wherever it goes

- Data Dryad
- Harvard Dataverse
- Center for Open Science
- Amazon S3
- ….

Share your Code with the World

- Second roadblock to reproducibility, lack of access to code
- as soon as you write code, put it in Github (You don’t have to publicize it right away)
- use a recognized Open Source License (http://opensource.org/)
- manage change to your code with intelligent, explanatory commits
- Organize each part of your pipeline into separate directories, or even repositories (you can use git subrepositories to organize them into a single unit)
- Include Documentation (Readme.md)
  - what it does
  - how to use it
  - software dependencies, installation

- Future you may be your happiest future user
- Ensures portability of your code to wherever you may roam
- Facilitiates portability of your code to different compute environments (OIT, DHTS, Amazon, etc.)
- Github repository url can be put in your publication (provided it exists before you submit the manuscript)
- Github forks represent adoption by the wider research community

Organize Code for Reproducibility

- use a fixed directory structure
- document your code liberally
- provide sensible defaults, usage statements, help when applicable
- design separate components to be (re)used in different contexts (yours and your future users)
- consider logging metadata to file/database (input files, output files, md5/sha1 hashes)

Containerize your Applications

- Docker.com
- Store Dockerfile with source code in Github
- Store/share docker images on registry.hub.docker.com
- use reciprocal references between Github and Docker Registry

- Docker containerized applications can use fixed directory structures
- No longer can sys admins tell you that you cannot have the latest version of X because other users need a previous version
- If you can run a docker container on one machine, you can run it on any docker host
- Arbitrary paths to data on host easily mapped to expected container directory structure
- Data packed volume containers can be used to automate process of downloading your publicly available data into the directory structure expected by the pipeline

A Working Example

https://github.com/dmlond/docker_bwa_aligner

Bwa alignment of P. falciparum sequence to reference

Images hosted on Docker Registry