Managing your research code

Helen Bailey & Phoebe Ayers
MIT Libraries
Data Management Services
IAP 2018
Code management can help…

- Others find, use, and build on your work
- Give you credit for your work
- Ensure your hard work isn’t lost -- to you or to others
- Document the work for funders & future commercialization
Welcome

- Goals of this workshop
- Things we’re not going to cover
- Discussion questions:
  - Why do you write code? What kind of code do you write?
  - What problems or questions do you have managing your code?
Workshop outline

- Research code landscape
- The lifecycle of your project
- Coding and collaborating
- Sharing and archiving
- Documentation
- Licensing (and MIT requirements)
- Citing code
- Resources
The landscape

- Writing code can be a core part of the research process!
- Reproducible science depends on code being available
- Landscape is changing fast:
  - Funder requirements to share code
  - Publisher support for sharing code
  - Open science = shared code & methods
Example funder policies

Grant data management plans - describe your data & how it will be shared

- **NSF guidance:**
  
  “Investigators and grantees are encouraged to share software and inventions created under the grant or otherwise make them or their products widely available and usable…. the Data Management Plan should describe the types of data, samples, physical collections, software, curriculum materials, and other materials to be produced …. What resources are needed to access or use the data? Examples are software or equipment.”

- **NASA guidance:**

  “Data are understood to include…descriptions of the software required to read and use the data, associated software documentation”
Example publisher policies

- **IEEE**: you can submit your code to CodeOcean and link to paper
- **ACM**: TOMACS and TOMS are publishing computational replication papers
- **Nature**: 
  - "Authors must make available upon request, to editors and reviewers, any previously unreported custom computer code used to generate results that are reported in the paper and central to its main claims. Any practical issues preventing code sharing will be evaluated by the editors who reserve the right to decline the paper if important code is unavailable. Upon publication, Nature Research Journals consider it best practice to release custom computer code…"
Workshop outline

- Research code landscape
- **The lifecycle of your project**
- Coding and collaborating
- Sharing and archiving
- Documentation
- Licensing (and MIT requirements)
- Citing code
- Resources
Code is part of a bigger picture

How does code fit into your research project?

- One-off scripts for data processing
- Data analysis scripts
- Visualization and presentation code
- Code as part of the research output (algorithms, libraries, executable programs)
- Other?
# Active project vs long-term maintenance

Needs and tools may be different!

<table>
<thead>
<tr>
<th>Sharing</th>
<th>Active project</th>
<th>Long-term maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Private, select individuals, team</td>
<td>Open source/public, available by request</td>
</tr>
<tr>
<td>Documentation</td>
<td>Focused on immediate needs (dependencies, installation steps, secrets)</td>
<td>All that plus general understanding (what does the code do, why certain decisions were made)</td>
</tr>
</tbody>
</table>
Talk to your neighbor

- How will your project/code be used by others?
- How long will the project run?
- Who needs to have access during the project?
- Who will need access once the project is completed?
- Who maintains the code during and after the project?
- Do other projects depend on this code?
Workshop outline

● Research code landscape
● The lifecycle of your project
● **Coding and collaborating**
● Sharing and archiving
● Documentation
● Licensing (and MIT requirements)
● Citing code
● Resources
Version control

- Essential to maintain history of changes to code and data
- For team projects, not just what changed but also who changed it
- Version control systems: git, subversion (SVN)
- Some software (R Studio, many IDEs) has git/svn integration built in

https://www.git-tower.com/blog/git-cheat-sheet/
# Sharing code-in-progress

Use an online repository!

<table>
<thead>
<tr>
<th></th>
<th>Regular Github</th>
<th>Enterprise Github</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Github.com</td>
<td>Github.mit.edu</td>
</tr>
<tr>
<td>Cost</td>
<td>Free and paid accounts</td>
<td>Free to MIT users</td>
</tr>
<tr>
<td>Privacy</td>
<td>Must have paid account for private repos, otherwise all repos are public</td>
<td>Unlimited private and public repos</td>
</tr>
<tr>
<td>Data storage</td>
<td>Hosted/stored on github.com servers</td>
<td>Hosted/stored on MIT servers</td>
</tr>
<tr>
<td>Non-MIT access</td>
<td>Yes</td>
<td>Yes, with Touchstone collaboration accounts</td>
</tr>
</tbody>
</table>
Workshop outline

- Research code landscape
- The lifecycle of your project
- Coding and collaborating
- **Sharing and archiving**
- Documentation
- Licensing (and MIT requirements)
- Citing code
- Resources
Where to put it long-term?

- For continued development:
  - GitHub, SourceForge, another public development repository
- For static downloads:
  - General data repository (Zenodo, Figshare, etc)
  - A data or software repository by language or discipline (eg CRAN)
  - Institutional data repository (eg dspace@mit)
  - As journal supplemental data
  - On your website....?
Choosing a repository

● What are your goals?
  ○ Further development with others
  ○ Access by those working in the scientific field
  ○ Clustered with similar packages in that language
  ○ Linked to your datasets
● Is it reliable? Has features you need? Is there a cost?
● Data repository list: https://www.re3data.org/
  ○ … not all have software
  ○ We can help!
Getting a DOI for your GitHub code

- An easy step to help you get credit & make code citable
- Archive a snapshot of your public GitHub repo by archiving it in Zenodo or Figshare (http://zenodo.org or http://figshare.com)
- Each new release is automatically archived
- Each new release gets a permanent DOI
- Can be linked to datasets

https://guides.github.com/activities/citable-code/
Sharing as paper supplemental material

- Can upload supporting files to most journals
- Typically small, static .zip files
- Useful if code won’t change and/or is only useful in context of paper
- Can be uploaded with datasets
- Special cases:
  - IEEE & SPIE support CodeOcean: executable, dynamic code platform
  - [https://codeocean.com/signup/ieee](https://codeocean.com/signup/ieee)
  - For computational reproducibility; supports C/C++, Fortran, Java, Julia, Lua, MATLAB, Octave, Perl, Python, Stata and R
  - Can also sign up with individual account
Backups vs long-term preservation

- Storage doesn’t ensure usability / access over the long term
  - Physical media degrades
  - Online repositories can change
  - Check backups for usability/fixity
- Consider the availability of libraries, data or other info needed
- Documentation important for long-term usability
  - What’s the story of this code?
Workshop outline

- Research code landscape
- The lifecycle of your project
- Coding and collaborating
- Sharing and archiving
- **Documentation**
- Licensing (and MIT requirements)
- Citing code
- Resources
Who needs to understand your code?

- You!
- Current teammates/coworkers
- Future users of your code
- Future maintainers, editors of your code
- Reviewers/publishers
- Funding agencies
What do they need to know about your code?

- Dependencies
- Installation and execution instructions
- API keys, database login, other secrets (be careful not to share publicly!)
- What is it for?
  - General overview, plus
  - What various files/components of the code do
- Overview of how the code is structured/organized
- Why certain key decisions were made (for example, using one analysis algorithm/method over another)
- Related data sets, publications, etc.
- Credit! (Who did what)
Where/how to document your code?

- Inline comments
- Documentation generators (Pydoc, Sphinx, Javadoc, etc.)
- Readme files
- Dependency management tools
- Methods sections and software papers
Containerization for reproducibility

Containers: virtual machines with specified environment configuration and versions of dependencies installed

- Allows others to recreate exact environment in which to run code
- Docker is one common container system
  - Has many common base containers already (including Jupyter notebook containers)
  - Publish containers to DockerHub
  - Well-documented
- Reprozip is a tool specifically designed to package research data, files, environment variables, etc. for reproducibility
  
  https://www.reprozip.org/
Workshop outline

- Research code landscape
- The lifecycle of your project
- Coding and collaborating
- Sharing and archiving
- Documentation
- **Licensing (and MIT requirements)**
- Citing code
- Resources
Should you license your code?

If you are publishing your code publicly, you should license it!

Types of open source licenses:

- MIT/OpenBSD: very permissive, attribution only
- GNU: requires licensed work and modifications to be published under same license
  - v.2.0: does not include patent clause
  - v.3.0: does include (confusing) patent clause

https://choosealicense.com/licenses/
MIT Technology Licensing Office

- Offers help with licensing and patent issues!
- Requests that any open source licensed code be disclosed to them for review
- **Definitely submit to TLO if your code is a deliverable for a sponsored project!**
- Note: MIT owns copyrighted software for any software created using MIT administered funds or facilities

Workshop outline

- Research code landscape
- The lifecycle of your project
- Coding and collaborating
- Sharing and archiving
- Documentation
- Licensing (and MIT requirements)
- Citing code
- Resources
Get credit: provide a citation

You can list a citation for your package in...

- a .CITATION file (eg in R)
- the Readme
- the paper
- the software landing page or website
- All of the above

Good citations…

- Make it more likely people will acknowledge you
- Make it easier to find the software
- Makes science more accurate
Citation elements

- **author(s)** in the order you would like them to be credited
- **title of the software package / code**
- **link to the location where the code can be downloaded or purchased**
- **DOI if you have created one, or other unique identifier** -- Force11 recommends that the identifier resolve to "*a persistent landing page that contains metadata and a link to the software itself, rather than directly to the source code files, repository, or executable.*"
- **version number**
- **release date**
- **license**
Give credit: cite others

- What packages or libraries did you use that were core to your analysis?
- Use the preferred citation. If there isn’t one:
  - include a download/project page pointer and name of the project, authors if available, version, license.
- *Also* cite a software paper if available
Software papers & software journals

Software papers...

- Primarily about the software and how it works
- Software itself is stored in a repository or data archive
- Gives you publication credit for your work
- May be a traditional journal (*The Astrophysical Journal, PLoS ONE* or a journal that only publishes articles about software (*The Journal of Open Source Software, The R Journal, SoftwareX*)
  - Software journal list: [https://www.software.ac.uk/which-journals-should-i-publish-my-software](https://www.software.ac.uk/which-journals-should-i-publish-my-software)
Software papers cont.

- May be short or long
- Gives you a citation that can be used in addition to citing the software itself
- Should tell you who, what, why, where and how
- Examples:
  - VanderPlas, “mst_clustering: Clustering via Euclidean Minimum Spanning Trees”
  - Barbeau et al, “ConfBuster: Open-Source Tools for Macrocycle Conformational Search and Analysis”
    - [https://openresearchsoftware.metajnl.com/articles/10.5334/jors.189/](https://openresearchsoftware.metajnl.com/articles/10.5334/jors.189/)
Resources

- Software guide: https://libguides.mit.edu/software
- Technology Licensing Office: https://tlo.mit.edu/
- MIT/BU Cyberlaw clinic: https://sites.bu.edu/tclc/
### Upcoming classes

- **Patents**: 4-5pm (now!)
- **Law & Technology: Know Your Rights Teach-in**
  - Th/Fri Jan 25 & 26, 2-6pm, E15-359
  - [https://sites.bu.edu/tclc/iap2018/](https://sites.bu.edu/tclc/iap2018/)
- **Data Management: File Organization**
  - Wed Jan 31, 4:00pm-5:00pm, 14N-132
  - [http://libcal.mit.edu/event/3769326](http://libcal.mit.edu/event/3769326)
- **Basics of Copyrights, Data and Software Intellectual Property**
  - Fri Feb 2, 10-11:30, 3-370
  - [https://calendar.mit.edu/event/basics_of_copyrights_data_and_software_intellectual_property](https://calendar.mit.edu/event/basics_of_copyrights_data_and_software_intellectual_property)

---

**What future workshops would be useful to you?**
We’re here to help!
data-management@mit.edu

http://libraries.mit.edu/data-management