Making Research Software More Robust

Morgan Taschuk[1] and Greg Wilson[2]

1 Ontario Institute for Cancer Research, Toronto, ON, Canada
2 Software Carpentry http://software-carpentry.org

Bioinformatics software is often written by a single person who is both developer and scientist, intent on producing insightful, novel results for their own projects. Other people may want to use their methods, but much of academic software is written in an ad hoc, exploratory style that is incompatible with reuse. Often, the end user will need to spend a considerable amount of time either reading code or in communication with the original developer in order to get the software running.

We present a set of guidelines to encourage code reuse that can be applied to any language, library, or environment for both open-source and closed-source software. By following these guidelines, bioinformaticians can smooth the transition from exploratory code to software that is intended for publication, production environments, and general reuse by the scientific community.

Read the paper and download the checklist: https://github.com/oicr-gsi/robust-paper

Introduce your software

Have a README

- ✓ Explain what the software does
- ✓ List dependencies
- ✓ Provide compile/installation instructions
- ✓ List input and output files
- ✓ State attributions and licensing

Bonus points: write high-quality documentation [1]

Ease installation woes

Use a build utility and package manager

- ✓ Use a build utility, even for scripts (e.g. Make)
- ✓ Document all dependencies in a machine-readable form
- ✓ Avoid depending on scripts and tools which are not available as packages (using pip, CRAN, CPAN, Maven…)

Version your releases

- ✓ Increment every time you release
- ✓ Print when supplying --version on the command line
- ✓ Include version number in output
- ✓ Deposit releases in a stable location so they are available in perpetuity

Eliminate hard-coded paths

- ✓ Set the names and locations of input and output files as command-line parameters
- ✓ Do not require users to navigate to a particular directory to do their work

Do not require root or special privileges

- ✓ Allow software to be installed in an arbitrary location
- ✓ Ask another person to try and build your software

Reuse software (within reason)

- ✓ Make sure that you really need the other program
- ✓ Check to make sure dependencies are available early in execution
- ✓ Use native functions for starting other processes

Bonus points: use a Docker container or similar to reduce installation woes [3]

Simplify execution

Print usage information on the command line

- ✓ Include:
  - ✓ syntax for running the program in GNU/POSIX format
  - ✓ a one line description
  - ✓ the most commonly used arguments, a description of each, and the default values
  - ✓ where to find more information
- ✓ Print to standard output
- ✓ Exit with an appropriate exit code

Allow configuration of useful parameters

- ✓ Choose reasonable defaults where they exist
- ✓ Set no defaults at all when there aren't any reasonable ones
- ✓ Echo all parameters and software versions to standard out or a log file
- ✓ Check that all input values are in a reasonable range near startup

Bonus points: conform to command-line conventions [2]

Increase user confidence

Produce identical results when given identical inputs

- ✓ For randomized algorithms:
  - ✓ allow the user to optionally provide the seed as an input parameter; or
  - ✓ make sure the acceptable tolerance is known and detailed in documentation and in tests

Include a small test set

- ✓ Tests should:
  - ✓ Exist!
  - ✓ Be easy to find
  - ✓ Be straightforward to run
  - ✓ Be simple to interpret

1. Karimzadeh M, Hoffman MM. Creating great documentation for bioinformatics software; 2016. http://hdl.handle.net/1807/73111
2. Seemann T. Ten recommendations for creating usable bioinformatics command line software. GigaScience. 2013;2(1):15. doi: 10.1186/2047-217X-2-15.
3. Docker – Build, Ship, and Run Any App, Anywhere. https://www.docker.com/

Funding for the Ontario Institute for Cancer Research is provided by the Government of Ontario.