Version control for scientific research

We live in an increasingly collaborative era, where the Internet enables distance collaboration almost trivially – not just with e-mail and videoconferencing, but with collaborative realtime document editing and networked transmission of data and analyses. These tools allow us to collectively leverage many resources to rapidly solve problems and ultimately accelerate scientific discovery. While these tools and technologies are fundamentally changing how we collaborate on science, there is still considerable room for improvement in how we are using them.

Programmers, and especially the world of open source software development, have developed a number of tools that enable easy distance collaboration and sharing of data and code. One of the most important is **version control** – tracking of changes and authorship – which, in programming, is largely used for writing and sharing code, tracking and annotating contributions, and resolving changes to the same section of a program. In recent years a wide variety of version control software has become readily and freely available, including Subversion, git, and Mercurial.

Motivated by much the same problems as programmers – how do I track changes and authorship, and resolve conflicting changes? – scientists have increasingly begun using these version control systems not only for software but for paper and grant writing, and even for tracking data and metadata. In particular, CTB and others have argued that version control should be a required "good practice" for all computational science in practice (G. Wilson et al. 2012). This increased interest in versioning for science has led to a realization that, collectively, we are lacking in workflows and practices that take advantage of version control. This is especially true for git and Mercurial, both decentralized version control systems that enable many different modes of collaboration.

One of us, KR, has just published a paper showing how "git can facilitate greater reproducibility and increased transparency in science." (Ram 2013). Occasioned in part by this first paper, a group of us is working to document and demonstrate the many uses to which we have put git. These include software development and dissemination, paper and grant writing, paper and grant feedback, contributions to Wiki-like community documentation sites, and sharing of data analysis "notebooks" and executable papers.

In tandem with git and Mercurial, two commercial Web sites have appeared, GitHub and BitBucket. These sites serve as central "hubs" through which code and other electronic artifacts can be communicated, shared, and collaborated upon.

We believe that KR's paper (Ram 2013) is a solid first step in showing how git and sites like GitHub can be used for versioning, collaboration, and feedback on research. Separately, CTB's group and others have explored git's use for publishing reproducible notebooks and enabling community documentation.

Git and Mercurial, and the sites that support public hosting of git and Mercurial repositories such as GitHub and BitBucket, do provide an excellent solution to the problem of posting code from published modeling and data analysis efforts. Rather than such code being hosted on university Web sites that languish unmaintained, or deposited in hard-to-find journal supplements (Schultheiss et al. 2011; Wren 2004), GitHub and BitBucket (among other sites) provide stable hosting for code. Moreover, since these sites are widely used outside the research community – most especially in the open source community – the likelihood that they will disappear with loss of funding body interest is low.

One of the most interesting potential uses for git and Mercurial is in supporting the "forking", or copying and modification, of data analysis pipelines, to enable field-wide reuse of the analysis pipeline provided in a published paper. Unlike more centralized systems such as Subversion, git and Mercurial both readily support diverging development of software starting from a common base by other groups. This kind of branching enables remixing of research software, while retaining all provenance information. Because projects can easily be made available for branching upon publication by using GitHub and BitBucket for post-publication development and dissemination, many barriers to remixing and reuse of software are significantly lowered.

Git and Mercurial, and the GitHub and BitBucket sites, are not perfect fits for the scientific process. While free public and private repositories are available to academics on both sites, neither site is archival, and so sites such as figshare or field-specific archives may be needed to provide a lasting home for published software and data.

We have only begun to understand and detail the many uses of distributed version control in conducting research, but already it offers many exciting opportunities to improve on existing workflows. We look forward to exploring this area in the future and building a real community of practice.

Our next steps in this area are to gather further examples of the use of version control for scientific research; develop tutorials, handbooks, and training materials; survey more scientific users on their current and planned practices; and ultimately publish additional works on how version control is being used and how it *could* be used to improve scientific practice and efficiency.

Literature Cited

Ram, Karthik. 2013. “git can facilitate greater reproducibility and increased transparency in science.” *Source Code in Medicine and Biology* xx (xx): xx.

Schultheiss, Sebastian J., Marc-Christian Münch, Gergana D. Andreeva, and Gunnar Rätsch. 2011. “Persistence and availability of Web services in computational biology.” Ed. Dongxiao Zhu. *PloS one* 6 (9) (jan): e24914. doi:10.1371/journal.pone.0024914. http://dx.plos.org/10.1371/journal.pone.0024914.

Wilson, Greg, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, et al. 2012. “Best Practices for Scientific Computing.” *Arxiv* (sep): 6.

Wren, Jonathan D. 2004. “404 not found: the stability and persistence of URLs published in MEDLINE.” *Bioinformatics (Oxford, England)* 20 (5) (mar): 668–72. doi:10.1093/bioinformatics/btg465. http://bioinformatics.oxfordjournals.org/content/20/5/668.abstract.