Introduction

Bioinformatics is a broad discipline in which one common denominator is the need to produce and/or use software that can be applied to biological data in different contexts. To enable and ensure the replicability and traceability of scientific claims, it is essential that the scientific publication, the corresponding datasets, and the data analysis are made publicly available. All software used for the analysis should be either carefully documented (e.g., for commercial software) or, better yet, openly shared and directly accessible to others. The rise of openly available software and source code alongside concomitant collaborative development is facilitated by the existence of several code repository services such as SourceForge, Bitbucket, GitLab, and GitHub, among others.

These resources are also essential for collaborative software projects because they enable the organization and sharing of programming tasks between different remote contributors. Here, we introduce the main features of GitHub, a popular web-based platform that offers a free and integrated environment for hosting the source code, documentation, and project-related web content for open-source projects. GitHub also offers paid plans for private repositories for individuals and businesses as well as free plans including private repositories for research and educational use as indicated by Perez et al., (2015)

Some of the recommendations outlined below are broadly applicable to repository hosting services. However, our main aim is to highlight specific GitHub features. We provide a set of recommendations that we believe will help the reader to take full advantage of GitHub's features for managing and promoting projects in bioinformatics as well as in many other research domains. The recommendations are ordered to reflect a typical development process: learning Git and GitHub basics, collaboration, use of branches and pull requests, labelling and tagging of code snapshots, tracking project bugs and enhancements using issues, and dissemination of the final results.

Body and Background

GitHub relies, at its core, on the well-known and open-source version control system Git, originally designed by Linus Torvalds for the development of the Linux kernel and now developed and maintained by the Git community. One reason for GitHub's success is that it offers more than a simple source code hosting service. It provides developers and researchers with a dynamic and collaborative environment, often referred to as a social coding platform that supports peer review, commenting, and discussion. A diverse range of efforts, ranging from individual to large bioinformatics projects, laboratory repositories, as well as global collaborations, have found GitHub to be a productive place to share code and ideas and to collaborate .

The backbone of GitHub is the distributed version control system Git. Every change, from fixing a typo to a complete redesign of the software, is tracked and uniquely identified. Although Git has a complex set of commands and can be used for rather complex operations, learning to apply the basics requires only a handful of new concepts and commands and will provide a solid ground to efficiently track code and related content for research projects. Many introductory and detailed tutorials are available (see Table 2 below for a few examples). In particular, we recommend A Quick Introduction to Version Control with Git and GitHub by Blischak et al., (2017).

The adoption and use of Git and Github within computer science education is growing in popularity. The motivation for this shift is strong: it combines a robust system for managing student coursework, sophisticated collaboration and communication tools for students and teaching staff, and an authentic experience of an important software engineering skill. Whilst previous literature has reported upon experience and benefits, there still exists a technical barrier to overcome in adopting Git and Github within an educational context. In response, both the community of teachers using Git/Github and the Github organisation itself have developed tool support to help solve the challenge of adoption, however these efforts are somewhat isolated and relatively unstudied. This work aims to provide an overview of these tools, identify the commonalities and differences, and develop a framework for comparison to assist teachers when looking for solutions for their own courses.

According to Davis et al., (2015), one of the most rapidly growing professional social networks is GitHub, an online space to share code. GitHub is based on free and open-source software called Git, a version control system used in many digital projects, from library websites to government data portals to scientific research. For projects that involve developing code and collaborating with others, Git is an invaluable tool; it also creates a backup system and structured documentation. In this article, we examine version control, the particulars of Git, the burgeoning social network of GitHub, and how Git can be an archival too

The main goal of this article is to expand on existing GitHub resources in order to share our recommended workflow for using GitHub Classroom as an educational tool and class management system. We begin in Section 2 with describing the practical and pedagogical benefits of using Github Classroom. In Section 3, we describe our experience in implementing GitHub Classroom in two educational settings—an introductory computational statistics (ICS) lab and a more advanced computational statistics (ACS) course. To allow instructors to more easily use GitHub Classroom, we describe the open source and publicly available tools and guides we have developed for using GitHub Classroom in Section 4.1. We have teacher-focused resources, which are targeted toward instructors (of all subjects) who wish to set up a GitHub Classroom, and student-focused resources, which can be distributed by instructors. Both resources provide visual guides to Git, GitHub, and GitHub Classroom for instructors and students who have never used version control before. The remainder of Section 4 discusses key aspects of our workflow for using GitHub Classroom, which are supplemented by our guides. We conclude the article with a brief summary and discussion in Section 5. We believe that our work, along with that of others, will help ease the larger statistics community into using Git and GitHub across the entire statistics curriculum Beckman et al., (2020).

2 Pedagogical and Practical Benefits of GitHub Classroom

Because there is a time investment associated with introducing GitHub and GitHub Classroom to students, it is worth discussing why instructors should implement GitHub Classroom to run a course, rather than using the default university course management system (CMS). An immediate advantage is for classes that have group projects. With GitHub Classroom, instructors can easily assign groups of students to teams and give each team their own GitHub repository within a GitHub Classroom. Students can then use Git and GitHub to collaborate on a project, just as they would in an academic or industry research project. Because teachers can see each student’s commit history, it is easy to see how each student contributed to the project. In addition, because instructors can easily apply for unlimited free private repositories associated with a GitHub Classroom, instructors do not have to limit the number of projects throughout a course for monetary reasons.

Even without group projects, however, GitHub Classroom has benefits over standard academic CMSs such as Blackboard, Sakai, and CoursePlus (Zagalsky et al. 2015). First, GitHub Classroom can be used to distribute and update course materials; GitHub can provide course structure without the instructor relying on their CMS. Students learn the most common Git commands such as clone, pull, and push, while staying up-to-date on course materials. Instructors can also encourage students to contribute to course materials (i.e., correcting mistakes in the lecture notes) through the Git and GitHub infrastructure (Zagalsky et al. 2015), keeping them engaged in the course.

It reduces the amount of work and number of chances for errors during the assignment creation workflow, relative to a standard CMS. We diagram how GitHub Classroom simplifies the assignment creation process in Figure 1. Instructors can maintain starter material for assignments on their local computer and can give students their own versions of the starter material with the push of a single button, as opposed to individually uploading each piece of the assignment to the CMS for students to download. Students can use Git to bring the full assignment onto their personal computer, without individually downloading each part of the assignment. We see in Figure 1 how individually downloading files can result in different file structures on each student's computer, which makes it far more likely that the resulting code will have errors or not be reproducible.

Conclusion

Previous work has shown that GitHub can be used for educational purposes across a range of subjects, class sizes, and instructor knowledge of GitHub (Zagalsky et al. 2015). The contribution of our work is to provide a concrete and easy-to-implement workflow for instructors who want to bring version control into their classroom. By using our recommended workflow, instructors will not only benefit their students by teaching them skills desired by potential employers but will also significantly cut down on the administrative work required to distribute, grade, and return assignments. By saving time that was formerly spent on administrative duties, instructors can spend more time working with students and updating course material.

Through our experiences, we show that the Git workflow can be used in both introductory and advanced courses and by instructors without previous GitHub experience. We have used student feedback to construct a separate guide to Git and GitHub for students, so that instructors unfamiliar with version control do not have to create their own teaching materials. Our hope is that our guides serve as a starting point for instructors to use GitHub, who will then modify and improve our workflows for different class settings.

References

1 . Beckman, M.D., Çetinkaya-Rundel, M., Horton, N.J., Rundel, C.W., Sullivan, A.J. and Tackett, M., 2020. Implementing version control with Git and GitHub as a learning objective in statistics and data science courses. *Journal of Statistics Education*, pp.1-35.

2. Davis, R.C., 2015. Git and GitHub for librarians. *Behavioral & Social Sciences Librarian*, *34*(3), pp.158-164.

3. Blischak, J.D., Davenport, E.R. and Wilson, G., 2016. A quick introduction to version control with Git and GitHub. *PLoS computational biology*, *12*(1), p.e1004668.

4. Fiksel, J., Jager, L.R., Hardin, J.S. and Taub, M.A., 2019. Using GitHub classroom to teach statistics. *Journal of Statistics Education*, *27*(2), pp.110-119.