Ten Simple Rules for taking advantage of GitHub in bioinformatics

Yasset Perez-Riverol, Rui Wang, Laurent Gatto, Timo Sachsenberg, Juan Antonio Vizcaíno1,\*

1 European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK.

\* Corresponding author: Dr. Juan Antonio Vizcaíno

European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD,UK. Tel: +44 1223 492 610; Fax: +44 1223 494 484; Email: juan@ebi.ac.uk.

**Short title:**

**Introduction**

Bioinformatics is a broad discipline in which the common denominator is the need to produce and/or use software that can be applied to biological data in different contexts. For instance software is routinely needed for the analysis, visualization [1] or storage of biological information. For enabling and ensuring the reproducibility of scientific claims, it is essential that, together with the scientific publication, the corresponding dataset/s are made publicly available to the scientific community [2]. In addition, all the software used for the analysis should be either well described (e.g. in case of commercial software) or openly shared, when possible [3,4]. At present the latter is becoming more common in the case of “home made” programming scripts or for bioinformatics open-source projects, where the source code is made freely available and can be reused by third parties [5]. The availability of the source code can be achieved thanks to the existence of several code repository-hosting services such as Sourceforge (<http://sourceforge.net/>), Google Code (<https://code.google.com/>) and GitHub (<https://github.com/>), among others. These resources are also essential for collaborative software projects, since they enable the organization and sharing of programming tasks between different contributors to the same project.

Different to other alternatives such as Google Code and Sourceforge, Github is not only a source code repository for programmers and software developers; it offers a complete suite a services and tools for version control of documents, source, data, and also a collaborative environment for research and international teams. The central feature of GitHub is the well-known and open-source version control system Git, designed and developed by Linus Torvals in 2005 to control and extend the Linux kernel development. Git has become the most widely adopted version control system; adopted by major companies like Google, Facebook, Twitter. GitHub is free to use for use for public projects and offers paid plans for private repositories, and the host of millions of open source projects. Here we aim to introduce the main features of GitHub and related tools and services, as a web-based platform, which offers a free and integrated environment for hosting the source code, the documentation and the web page of open source projects. We will provide a set of recommendations for taking full advantage of the core features of GitHub to facilitate and manage the work on a given project and increase its profile to the scientific community. The manuscript is timely due to the announcement (March 2015) of the closure of another widely used service, Google Code. Therefore, many of the users of Google Code will therefore migrate to GitHub in the coming months.

**Rules**

**Rule 1. Structure your research projects: organization, repositories, and teams**

GitHub is structured around three major concepts: organisations, repositories and contributors. The organizations simplify management of group-owned repositories (for example: your company's code), expand on our permissions system, and help focus your GitHub workflow for business and large open source projects.

**Rule 2. Organize efficiently the work in your team involving external contributors.**

Issue Tracker

Code branches

Continuous integration

**Rule 3. Promote your project/s in the scientific community**

Social activity of GitHub

Code snippets (Gist)

**Rule 4. Use GitHub web pages to make documentation attractive to your users and other possible future collaborators** (use Git web pages)

**Rule 5. Always link and highlight your deposited source code in publications**

As already mentioned in the introduction, reproducibility of the scientific claims should be enabled by providing openly the software and the datasets that are used in a particular study described in a publication. You should always highlight as much as possible in your publications that your code is freely available in GitHub, together with any other relevant piece of information that you may have deposited.

In our experience, this openness definitely increases your chances of getting the paper accepted for publication. On one hand, journal editors and reviewers have the opportunity to reproduce your findings during the manuscript review process and realize that you have nothing to hide. On the other, once the paper is published, the same can be done by any member of the scientific community, which can increase your opportunities for further discussion and collaboration.

**Rule 6. Build up your scientific career**

To move ahead in your scientific career or simply to find a new job, it is very convenient to show and demonstrate that you have contributed to a given project and at which level. In GitHub it is easy to show the level of contribution of individuals to a particular code base. It is then possible for third parties to appreciate whether one given person has been one of the main contributors or does only sporadic contributions, or to which projects someone is contributing in a regular basis. This can be very useful information for a prospective employer.

Another complementary way to build up your CV is to use GitHub to contribute to the increasingly used “altmetrics” methods to capture the impact of your work. In scientific publishing, altmetrics are defined as non-traditional metrics that have been proposed as an alternative the traditional citation impact metrics [23538811;25423184]. Altmetrics cover not just citation counts, but also other aspects of the impact of a work, such as how many data and knowledge bases refer to it, article views, downloads, or mentions in social and news media. One way to achieve this is to use the service “ImpactStory” (<http://www.impactstory.org/>). ImpactStory is an open source, non-profit organisation that provides web-based tool that provides altmetrics for blog posts, datasets, reviewer activity and software, among others. Users can link their ImpactStory profile to their GitHub account, displaying information such as the popularity of the project, or the number of *forks* (repository copies) that other people have made of your code. At present, ImpactStory is a subscription-based service. However, users can try the service for free for 30 days.

**Rule 7. Make your code easily citable by others in publications**

In research, it is always a good practise to be able to cite items using permanent and unambiguous identifiers. It routinely happens for proteins, genes, metabolites, different types of datasets, or published references, among many other items. Digital Object Identifiers (DOIs) have been used for many years as unique and unambiguous identifiers for enabling the citation of scientific publications. More recently, a trend has started to produce DOIs for other types of scientific outputs such as biological datasets [24727771] or training materials. The main motivation behind is to give scientists a better credit for their work [19587644], enabling at the same time a better way to cite and track it. Following this trend, GitHub now enables the use of DOIs to cite the code deposited, using the data archiving tool Zenodo (<https://zenodo.org/>). The procedure is very simple and it is explained in detail here (<https://guides.github.com/activities/citable-code/>). By default, Zenodo takes an archive of your GitHub repository each time you create a new release. Before Zenodo can issue a DOI for your repository, you will need to provide some metadata information about the archived GitHub repository. Once the DOI has been assigned, apart from using it in your CV or add it to information resources such as Europe PubMed Central [25378340].

**Rule 8. Explore new ideas and discover the power of open source projects**

One of the jobs of a scientist is to read routinely the published literature available. Analogously, one of the jobs of programmers should be revise publicly available projects and code that can be interesting for his/her interests. GitHub makes this possible

Explore projects

Fork projects

**Rule 9. Be social.** Follow projects, people, projects, and issues.

You should try to learn as much as possible from your peers and colleagues and be updated with the developments of a project you are interested in. GitHub enables functionality that is already common is commonly used Social Media platforms such as Facebook or LinkedIn.

**Rule 10.** **Keep going!**

If you have never used GitHub, especially if you don’t have experience in using a versioning control system, there is the typical learning curve that can be applied to everything new someone has to do. However, in time, you will realize of all the time that you have saved by using GitHub.

**Conclusions**

To summarize, you can benefit for using GitHub in many different ways.

**Acknowledgements**

Y.P.R is supported by the BBSRC ‘PROCESS’ grant [reference BB/K01997X/1]. R.W. is supported by the BBSRC ‘Quantitative Proteomics’ grant [reference BB/I00095X/1]. J.A.V. is supported by the Wellcome Trust [grant number WT101477MA].

References

1. Wang R, Perez-Riverol Y, Hermjakob H, Vizcaino JA (2015) Open source libraries and frameworks for biological data visualisation: a guide for developers. Proteomics 15: 1356-1374.

2. Goodman A, Pepe A, Blocker AW, Borgman CL, Cranmer K, et al. (2014) Ten simple rules for the care and feeding of scientific data. PLoS computational biology 10: e1003542.

3. Vihinen M (2015) No more hidden solutions in bioinformatics. Nature 521: 261.

4. Osborne JM, Bernabeu MO, Bruna M, Calderhead B, Cooper J, et al. (2014) Ten simple rules for effective computational research. PLoS computational biology 10: e1003506.

5. Leprevost Fda V, Barbosa VC, Francisco EL, Perez-Riverol Y, Carvalho PC (2014) On best practices in the development of bioinformatics software. Front Genet 5: 199.

**Figures.**

**Figure 1.**