Good Enough Practices in Scientific Computing: A Learning Module for Researchers

Tomasz Zieliński [1], **Andres Romanowski** [1,2], **Emma Wilson** [3], **Felicity Anderson** [1,4], **Elvina Gountouna** [5,6], **Matthias Mimault** [7], **Alison Meynert** [5], and **Edward W. J. Wallace** [1]

1 School of Biological Sciences, The University of Edinburgh **2** Plant-Environment Signaling Group, Dept. of Biology, Faculty of Science, Utrecht University, Utrecht, 3584 CH, The Netherlands **3** Centre for Clinical Brain Sciences, The University of Edinburgh **4** EPCC, The University of Edinburgh **5** MRC Human Genetics Unit, MRC Institute of Genetics and Cancer, The University of Edinburgh **6** Usher Institute, The University of Edinburgh **7** The James Hutton Institute, Dundee

Summary

We present a half-day learning module targeted at a broad audience of researchers who want to learn how to be more efficient and effective in their data analysis and computing, whatever their career stage. The module teaches "good enough practices" that are near-universally useful for researchers who use computers in their work. These practices encompass data management, software and programming, collaborating with colleagues, organizing projects, keeping track of changes, and writing manuscripts. Good enough practices rely on a shared set of principles that span these areas: planning, modular organization, names, and documentation. The lesson is in The Carpentries format and the materials are open-source and hosted on GitHub by The Carpentries Lab. The lesson is visible at https://carpentries-lab.github.io/good-enough-practices/.

Statement of Need

This lesson addresses good practices for data analysis, computing, and collaboration, that are broadly useful for researchers. The underlying principles of planning, modular organization, good names, and documentation, are important for both novices and experts who take computational approaches. This learning module helps novices to learn these good practices in their own right, empowering them to learn other computational skills. For example, novice programmers can struggle with poor planning, poorly organized files and projects, unclear names for variables, documenting and describing their code. Teaching good practices separately means that programming lessons can focus on programming concepts and syntax. Making good practices explicit also helps more advanced researchers to collaborate with others, and to explicitly teach good practice to novices.

Learning objectives, design, and experience

The learning objectives span several episodes:

- Introduction: How can we use computers more effectively in a scientific project?
- Data Management: What is data management? How should I back up my data? How should I share my data?
- Code and Software: What is research code and software? What can you do to make code usable and reusable?

- Collaboration: What do collaborators need to know to contribute to my project? How can documentation make my project more efficient?
- Project Organization: How should I name my files? How does folder organization help me?
- Keeping Track of Changes: How do I make changes to a project without losing or breaking things? Why does GitHub exist?
- Manuscripts: How do I write a collaborative paper?

The approach is to emphasise a set of shared principles across the episodes:

- Planning how to work is essential, and any plan that you and your collaborators can stick to is better than no plan.
- Organizing your data, code, and projects into coherent modules makes them easier to understand and to re-use.
- Giving good names to your files, folders, and functions, makes them easy to find, to understand, and to use.
- Explicitly documenting everything that you and your collaborators need to know in the future communicates your plans and organization.

The instructional design is for an instructor-led half-day workshop module, arranged in episodes with clear objectives. Short format courses can be an effective way to encourage long-term improvement in practices (Jordan et al., 2017), and our module development drew on The Carpentries' lesson development materials (Backhaus et al., 2024) (although in its previous form as the curriculum development handbook (Becker & Michonneau, 2021)). Each episode contains interactive materials in the form of challenges or discussions that stimulate peer-to-peer learning. A shared collaborative document is used by instructor and learners, which itself exemplifies good practices. A template for the collaborative document is included in the learning materials. Instructor notes give more detailed suggestions. The learning materials are hosted online and provide further suggestions and links for self-study.

Good Enough Practices has been delivered to and evaluated by PhD and post-doctoral researchers over several years, both in-person and online via video call. In particular, we have delivered to first-year PhD students in multiple training programs at The University of Edinburgh. Feedback on the workshop is overwhelmingly positive, and different groups of learners learn different things from it. We have found that the discussion format, along with asking learners what they seek to gain from the lesson, allows instructors to place emphasis according to the level of the learners. For example, this can be the first time that a learner hears about README files. Alternatively, it can be the time where one learner says "this is very nice but you don't really need to write a README file do you" and other learners tell persuasive anecdotes arguing that, yes, you do.

We recommend the workshop for first-year PhD students, for researchers who are moving through collecting data towards larger-scale data analysis, and for those who have started to write code and need help organising it. We particularly recommend this workshop as a preliminary before learning about version control with Git (Gonzalez et al., 2019). For researchers in biomedical sciences, it is a good preliminary before Project Management and Organization for Genomics (Koziar et al., 2023), which gives practical applications of the principles outlined in this workshop.

How the lesson came to be

The lesson is inspired by and based on the paper, Good Enough Practices in Scientific Computing (Wilson et al., 2017): "a set of good computing practices that every researcher can adopt, regardless of their current level of computational skill".

After many years teaching computational skills via Data Carpentry, Software Carpentry,

and mentoring research students, we came across the same problems repeatedly: learners get stuck on names, concepts, and organization. Learners may come to programming lessons with a vague idea that "they should learn Python" (or whatever), but without a clear idea as to why they need to. We noticed a particular gap in the Software Carpentry Git lesson (Gonzalez et al., 2019), where learners reported that the lesson told them how to use Git but did not explain what it is for. We aimed to fill this gap by adapting the Good Enough Practices paper (Wilson et al., 2017) into a workshop, initially as PowerPoint slides. Within the UKRI-funded Ed-DaSH project, we turned the material into a Carpentries-style lesson, which required distilling key messages and concepts using less text, while adding more interactive material. After 3 years of iterative improvements based on learner and instructor feedback, we find the material to be very effective. One gratifying piece of feedback was from a PhD program administrator who started writing README files for all of their folders after hosting an online edition of the lesson.

Acknowledgements

References

Backhaus, A., Becker, E., Bellini Saibene, Y., Brooks-Kieffer, J., Companjen, B., Curty, R., Davey, R., Hertweck, K., Hodges, T., Kamvar, Z., Khan, B. J., Kohrt, F., Kuzak, M., McClatchy, S., Michonneau, F., Nenadic, A., Stevens, S., Takemon, Y., Trizna, M., … Word, K. (2024). *The Carpentries collaborative lesson development training.* Zenodo. https://doi.org/10.5281/zenodo.14360351

Becker, E., & Michonneau, F. (2021). *The Carpentries curriculum development handbook.* https://cdh.carpentries.org

Gonzalez, I., Huang, D., Hejazi, N., Koziar, K., & Munk, M. (Eds.). (2019). *Software Carpentry: Version control with Git.* Zenodo. https://doi.org/10.5281/zenodo.3264950

Jordan, K. L., Marwick, B., Weaver, B., Zimmerman, N., Williams, J., Teal, T., Becker, E., Duckles, J., Duckles, B., & Wickes, E. (2017). *Analysis of The Carpentries' long-term feedback survey.* https://doi.org/10.5281/zenodo.1039944

Koziar, K., Hejazi, N., Munk, M., & Gruber, S. (Eds.). (2023). *Data Carpentry: Genomics organization lesson.* Zenodo. https://doi.org/10.5281/zenodo.7908089

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLoS Computational Biology*, *13*(6), e1005510. https://doi.org/10.1371/journal.pcbi.1005510