Electronic Communications of the EASST
Volume 83 Year 2025

deRSE24 - Selected Contributions of the 4th Conference for Research Software Engineering in Germany

Edited by: Jan Bernoth, Florian Goth, Anna-Lena Lamprecht and Jan Linxweiler

Teaching Reproducible Data Analysis for HPC Users – The Snakemake Teaching Alliance

Malte Petersen, Sören Lukas Hellmann, Florian Böker, Fabian Brand, Aasish Kumar Sharma, Martin Leandro Paleico, Johannes Köster, Christian Meesters

DOI: 10.14279/eceasst.v83.2600

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Teaching Reproducible Data Analysis for HPC Users – The Snakemake HPC Teaching Alliance

Malte Petersen[1], **Sören Lukas Hellmann**[2,3], **Florian Böker**[1], **Fabian Brand**[1], **Aasish Kumar**[4], **Martin Leandro Paleico**[4], **Johannes Köster**[5], **Christian Meesters**[6]

[1]High Performance Computing & Analytics Lab, University of Bonn, Germany; [2]Faculty of Biology, Nucleic Acids Core Facility, Johannes Gutenberg University Mainz, Germany; [3]Institute for Quantitative and Computational Biosciences, Johannes Gutenberg University Mainz, Germany; [4] Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen, Germany; [5]University Hospital Essen, Germany; [6]NHR Süd-West, Johannes Gutenberg University Mainz, Germany

Abstract: In this article, we introduce the Snakemake HPC Teaching Alliance, a collaborative effort within the Snakemake project aimed at teaching reproducible data analysis to high-performance computing (HPC) users. We address the challenges encountered by researchers new to HPC workflows, aiming to facilitate seamless integration of distributed cluster computing using Snakemake's batch system support. Our alliance creates educational resources to enable data analysts at every carrier level carrying out reproducible data analysis at large scale, foster HPC accessibility and standardization in bioinformatic workflow solutions aligned with Open Science principles.

Keywords: Snakemake, HPC, Reproducible Data Analysis, Open Science, Teaching, Education

1 Introduction

Despite the immense computing capacity provided by HPC clusters, and the substantial investment made by governments to obtain these resources, it remains unfamiliar territory for many life science researchers [PTB+18, CT12, LDO+13]. The primary barriers include the opacity of usage procedures, the bureaucratic hurdles associated with accessing HPC resources, and a notable lack of accessible support systems as compared to formulated ideals [KCN+20, LDO+13].

The intricate process of gaining access to HPC systems, coupled with the convoluted nature of usage procedures, has led to a situation where life science researchers (and presumably researchers in other fields with no inherent exposure to coding and HPC Systems) seek alternatives. In response to this demand, institutions may resort to establishing additional localized infrastructures, resulting in a fragmented and non-standardized computing environment with numerous duplicated resources. The consequence is a landscape marked by redundancy, inefficiency, and a lack of standardization in practice, all of which hinder the realization of the full potential of HPC in life science research. – However, the authors acknowledge that some institutions have established dedicated compute resources and support, i. e.[CT12, LDO+13, CGF+20].

To bridge this gap between community requirements, a pivotal step involves the adoption of contemporary workflow management systems that seamlessly integrate with HPC batch systems and support remote file management for research data management support. We therefore founded the ”Snakemake HPC Teaching Alliance” in late 2023, an open group of volunteers dedicated to creating and promoting teaching material about the Snakemake Workflow Management System [MJL+21]. Using the native batch system support built into the Snakemake, we strive to overcome conventional challenges and enable users to tap into the full capabilities of distributed cluster computing.

The Snakemake workflow management system is a powerful tool designed for creating reproducible and scalable data analyses. Utilizing a human-readable, Python-based language, Snakemake makes it easy to create new and update existing analysis workflows. With its careful design choices – like the plugin system for extending functionality – it can seamlessly scale workflow deployment from a laptop to servers, clusters, grids, and cloud environments; all without modifying the workflow definitions, and while automating deployment of required software. As a final workflow output, Snakemake generates interactive visual reports that encapsulate results and data provenance, and can easily be shared with collaborators.

To illustrate what can be accomplished using the early development of our course material, we want to point out that considerable work has been accomplished previously by students supervised by our lecturers [PAR+22, OAR+23, AKA+24]. We note that the need to acquire working knowledge about workflow systems is recognized by the community of researchers striving for reproducible data analysis as a whole [MNB+17] and by the latest release of the HPC Certification Forum’s knowledge tree in particular [KFMG20].

Here, we present our advancements and extend an invitation to engage with the Snakemake developer community. Collaboratively, we can contribute to making workflow solutions more accessible and in alignment with Open Science goals. At the time of writing, the HPC Carpentry project (www.hpc-carpentry.org) hosted outdated material for executing Snakemake on HPC cluster. We got informed, that an overhaul is in progress and will be in contact with the HPC carpentry team. It is our aim not to compete, but rather to cooperate with other teaching initiatives.

2 Course Structure — Conceptualization

Whilst teaching online seems on the rise, little or no evidence is present to assess its efficiency. A meta-analysis pre-dating the corona pandemic states that online courses only exhibit minor benefits, if at all [AGK+11]. And a post pandemic survey found that students are rather dissatisfied with their online learning achievements [TAO+23] or rather miss their “classroom classes” [BPS+23]. Still, video-based courses *can* be effective [SWCR18]. The key to improving courses is engaging participants with hands on tasks and challenges, and it is estimated that “. . . the learning benefit from extra doing . . . to be more than six times that of extra watching or reading.” [KKJ+15]. This has been known in the computational community for a long time, with live coding being the most effective approach to teach programming [RKPV14, HCA+16, RPHH18]. However, designing, testing, and refining scripts that work on otherwise unknown clusters, is a substantial overhead when planning online tutorials.

We focus on delivering in-person, on-site courses using our teaching materials, integrating interactive elements such as code cloze texts, Parson problems [PH06, EMMG15, Eri16], and live coding with real-time execution to enhance learning outcomes. However, prior research indicates that students often retain little from their first programming courses [UTM+13]. Given that many of our participants may be first-time HPC users with limited programming experience, we emphasize strategies to improve teaching effectiveness. Specifically, we structure our approach around four distinct skills taught in a deliberate sequence [XLN+19]:

	semantics of code	patterns related to goals
reading	1. read code and predict behaviour	3. recognize templates and their uses
writing	2. write correct syntax	4. use templates to meet goals

Table 1: Summarizing the learning process to enable workflow users and programmers.

The Snakemake HPC Teaching Alliance focuses on three main audiences:

1. "Users" – data analysts who would like to use public Snakemake workflows
2. "Creators" – scientists who want to create and publish novel Snakemake workflows (we hence focus on possible future contributors of workflows to the "Snakemake Workflow Catalog")
3. "Administrators" – HPC teams striving to provide a better support for data analytics based on Snakemake

The course has been conducted several times already. We take feedback seriously and try to assess various information items, i. e. the computational skill level of our participants, their career level and Open Science commitment. Additionally, we ask our participants to provide feedback on the quality of a course: After every course, a link to an online survey is provided to all participants. All feedback is considered in our release cycle.

2.1 Course Component: "Users"

Many Snakemake workflows have been published, with the workflow GitHub repository "The Snakemake Workflow Catalog" collecting many (at the time of writing: > 240 standardized workflows, and ca. 2800 workflows contributed by the Snakemake community, see https://snakemake.github.io/snakemake-workflow-catalog/). In theory, every workflow can be downloaded and, after filling in the configuration parameters (e. g. file paths and application-specific parameters), executed on a cluster. Yet, most workflows will need modification to a varying degree to be executed in different environments effectively. We therefore focus in our training on

- selecting and deploying suitable workflows,
- executing a sample workflow on a cluster head node (this sample workflow is not computationally intensive.),

- managing software environments using Snakemake, Bioconda [TGD+18], HPC module files [Joh91], and containerized software [KCB+21],
- running a sample workflow while optimizing resource usage, including CPUs, memory allocation, and file system efficiency,
- incorporating Snakemake wrappers from the Snakemake wrapper repository to simplify the execution of complex software,
- introducing “group” jobs to improve resource management and efficiency,
- generating publication-ready reports.

Furthermore, solving frequent issues is taught;

- identifying those jobs which merit execution on a cluster and those which can easily be executed on a login or head node. These are steps which are not computationally intensive, e. g. workflow components which download reference data or performing a simple plot. Our participants are then taught how to exclude such steps and execute the sample workflow, now differentiating between local and batch job execution.
- identifying components which might lead to issues and adding logging information to the sample workflow
- identifying intermediate files – those which do no constitute initial data or final results – and labelling them as “temporary”. Snakemake will automatically delete those files if no longer needed. Thereby, we relieve users partially from future data management tasks.
- selecting files that need to be protected (prevented from manual deletion) or cached (e. g. such that subsequent workflows may re-use indexed references, which in turn can save compute time)

Particularly, workflow profiles for HPC clusters are still missing for many public workflows. We teach participants how to write such profiles for a particular workflow, and how to contribute changed code or workflow profiles back to the original developers of a particular workflow. This combination of skills enables participants to become part of the greater Snakemake community.

Participants are provided with configurations for Conda software deployment (so-called `.condarc` files) tailored for ”their” cluster. This includes, if needed, the use of http-proxies. Additionally, Snakemake is capable of adjusting to file system latency and able to stage-in input files. It thereby delivers a built-in remedy mitigating I/O issues (i. e. massive random access). All participants get some basic background information and a profile configuration file for “their” cluster.

2.2 Course Component: ”Creators”

This part addresses the need of data analysts to plan and code their own HPC capable analysis workflow. In addition to the course items outlined above, workflow creators learn the ins and outs of the Snakemake workflow manager.

We teach best practices in designing workflows, i. e. avoiding code duplications, validating inputs and configurations, and generating publication ready figures and reports. Since many groups in other research fields (high-energy physics, condensed matter physics, economics) resort to designing reproducible data analysis workflows in Snakemake, we provide a short intro into the example from bioinformatics. Future versions of our material might include the selection of examples from other scientific disciplines.

Snakemake wrappers (https://snakemake-wrappers.readthedocs.io/en/stable/) improve the reusability of workflows by standardizing the interface to many common tools and programs. They provide a great starting point for workflow creators to integrate applications that might be challenging to work with, thereby reducing the time it takes to write the first successful workflow considerably. Hence, introducing the use of Snakemake wrappers plays a considerable part in our course. However, an introduction into wrapper programming is a further on the horizon as basic Python knowledge is a prerequisite.

2.3 Course Component: “Administrators”

HPC teams interested in supporting “their” Snakemake users are shown what global settings can be applied and how they influence the behaviour of Snakemake on a cluster. Administrators can manage these settings like any other “dotfile” they provide and update according to either updates in their cluster configuration or along with updates in Snakemake itself.

A common situation where HPC administrators can benefit from knowledge of Snakemake is in the typical bioinformatics workload with many concurrent sequence alignment jobs. These workloads can cause massive random I/O operations onto a parallel filesystem (perceived by the parallel file system, if the reference index is bigger than the file system cache), HPC service providers may not want to rely on bioinformatic users staging their reference files onto node-local file systems themselves. A Snakemake configuration ensures that the Snakemake workflow manager will correctly handle these cases. It is ensured that such a global configuration does not interfere with other popular workflow managers like Nextflow [DCF+17].

As this “course set” for HPC service providers is rather short, it can be taught after completing one of the previous courses.

3 Practical Considerations

To facilitate teaching this course at different HPC centres with varying setups, we designed a questionnaire for HPC service teams. It is to be sent before any course, if cluster settings are unknown to the lecturer. Parameters asked for include an accessible path during the course (e. g. a workspace), quota, whether downloads need a web proxy, etc. The slide sets are based on LaTeX and the obtained metadata is used to render them using the Jinja 2 Python package [noa]. This includes the option to tailor a slide set for a more appealing design. Guidelines for lecturers, that can help them to get familiar with the material, are also included.

Many novices struggle to predict the output of short pieces of code and to select the correct completion of the code from a set of possibilities when told what it was supposed to do [LAF+04, LFT09]. We therefore explicitly instruct participants how to debug because we know good debuggers to be good programmers [HC18].

The three separate courses take about 1 day or 8 work hours for the “Users” course, 2 days for the “Creators” course and 30 to 45 min for the “Administrators”, including time for breaks.

4 Preliminary Feedback

The course has been taught twice. Additionally, precursor courses were taught as a one-week practical course for new students and included basic shell lessons, how to work remotely and carrying out a realistic workflow in the end. The first time this particular course was held, was as an NHR course in Dresden, with 17 participants. For future courses, we provide an online survey form to gather feedback systematically. We ask for the background of participants and the perception of the course quality in detail. Whenever a course will be conducted, lecturers will present a QR code directing participants to an online survey. This ensures systematic feedback over an extended period of time. Results will be summarized and provided to lecturers (upon request). This effort will allow us to improve our contents in the future, guided by participant feedback.

Preliminary results of the two mentioned courses are promising: Participants were pleased with the level of the course (13 of 14 received feedback forms; except 1 computational scientist, who found it too easy). Likewise, 8 of 14 gave an overall grade of “very good”, 6 of “good” (on a scale of “very good”, “good”, “fair”, “poor”, “very poor”. Furthermore, the timeframe of two days for the creator part seems appropriate. The evaluation of the curricular course encompassed an introduction into shell and python programming, too. It is therefore not comparable. However, the students considered the course to be of mark 1.9 on a scale from 1 (very good) to 6 (miserable).

We are aware of the preliminary nature of the reported feedback. The future collection of feedback along with the check for contributions to the Snakemake ecosystem from participants will allow for a more detailed evaluation, than just the perceived quality of the course.

5 Community Building

Beyond individual skill development, fostering a community around reproducible workflows in HPC is essential to sustaining long-term engagement and knowledge-sharing. Our alliance aims to create an ecosystem where lecturers can connect, exchange ideas, and share challenges and solutions. We want to educate data analysts to contribute their solutions back to the “Snakemake Workflow Catalog” (https://snakemake.github.io/snakemake-workflow-catalog/) for broader community use. Despite the availability of scalable workflow management systems, many new data analysts still use semi-manual methods. Our project integrates with the Snakemake project and by offering an open HPC trainer community we exchange ideas, and share challenges and solutions to improve the existing material.

6 Outlook

We intend to explore the feature space of Snakemake even further and will add further modules to the teaching material, which can be selected based on the expected skill level of the participants and site-specific needs. Two examples include

- A recent study among cell-biologist has found that data management remains a huge challenge [SHM+22]. Extending the teaching material with automated data management examples like automated gathering of metadata or providing examples for the iRODS data manager [XRC+17] and other systems is in preparation.

- Snakemake allows defining workflows that are dynamically updated at runtime. This is particularly beneficial for compute intensive tasks like fiducial tracking after deconvolution in image processing. Images or image stacks can be transferred from a microscopy laboratory to an HPC cluster, Snakemake triggers analysis jobs, reports are generated and transferred back. It also allows for running of partial workflows, like quality assurance and mapping tasks a sequencing facility might want to run for its clients, who can upload their files, trigger jobs and receive data for further analysis. For both scenarios, examples are being developed and will be included in the teaching material.

Being a part of the Snakemake development team helps us to include new features into the teaching material soon after a release. For instance, this will include the software deployment based on the European Environment for Scientific Software Installations (EESSI) [DHH+23] which is currently in development.

It is our aim to provide all our material for new

a) contributors to the Snakemake Ecosystem to

b) foster reproducible data analysis.

Third-party lecturers will find a detailed instruction set and setup, which is already partially developed. With the help of Jinja 2 [noa] we enable lecturers to render slide sets with their institutional layouts. Our long-term goal is to provide slide sets and materials which are ready to deploy.

7 Availability

All material outlined in this article is available at
https://github.com/snakemake/snakemake-hpc-teaching-material.
The entire material is released under the Creative Common Licence CC-BY-SA version 4.0 https://creativecommons.org/licenses/by-sa/4.0/.

Acknowledgements: C.M. thanks Prof. Julian Kunkel (GWDG) for his generous support in time and staff hours. S.L.H. and C.M. thank Dr. Christiane Kraemer for her support and feedback during the early stages of material development and the opportunity for testing it in her curricular courses.

Bibliography

[AGK$^+$11] R. Aggarwal, N. Gupte, N. Kass, H. Taylor, J. Ali, A. Bhan, A. Aggarwal, S. D. Sisson, S. Kanchanaraksa, J. McKenzie-White, J. McGready, P. Miotti, R. C. Bollinger. A Comparison of Online versus On-site Training in Health Research Methodology: A Randomized Study. *BMC Medical Education* 11(1):37, Dec. 2011.
doi:10.1186/1472-6920-11-37
https://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-11-37

[AKA$^+$24] S. Abdalfattah, C. Knorz, A. Ayoobi, E. A. Omer, M. Rosellini, M. Riedl, C. Meesters, T. Efferth. Identification of Antagonistic Action of Pyrrolizidine Alkaloids in Muscarinic Acetylcholine Receptor M1 by Computational Target Prediction Analysis. *Pharmaceuticals* 17(1):80, Jan. 2024.
doi:10.3390/ph17010080
https://www.mdpi.com/1424-8247/17/1/80

[BPS$^+$23] P. Baby, A. Philip, A. E. Simon, A. M. Sibi, A. M. Shaju, A. M. Abraham, B. V. Kathyayani. Undergraduate Nursing Students' Experiences of Attending Online Classes During the COVID-19 Pandemic. *Education in the Health Professions* 6(1):42–45, Jan. 2023.
doi:10.4103/EHP.EHP$_2$5$_2$2
https://journals.lww.com/10.4103/EHP.EHP_25_22

[CGF$^+$20] T. Castrignanò, S. Gioiosa, T. Flati, M. Cestari, E. Picardi, M. Chiara, M. Fratelli, S. Amente, M. Cirilli, M. A. Tangaro, G. Chillemi, G. Pesole, F. Zambelli. ELIXIR-IT HPC@CINECA: high performance computing resources for the bioinformatics community. *BMC Bioinformatics* 21(S10):352, Aug. 2020.
doi:10.1186/s12859-020-03565-8
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03565-8

[CT12] L. C. Crosswell, J. M. Thornton. ELIXIR: a distributed infrastructure for European biological data. *Trends in Biotechnology* 30(5):241–242, May 2012.
doi:10.1016/j.tibtech.2012.02.002
https://linkinghub.elsevier.com/retrieve/pii/S0167779912000170

[DCF$^+$17] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, C. Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology* 35(4):316–319, Apr. 2017.
doi:10.1038/nbt.3820
https://www.nature.com/articles/nbt.3820

[DHH$^+$23] B. Dröge, V. Holanda Rusu, K. Hoste, C. Van Leeuwen, A. O'Cais, T. Röblitz. EESSI: A cross-platform ready-to-use optimised scientific software stack. *Software: Practice and Experience* 53(1):176–210, Jan. 2023.
doi:10.1002/spe.3075
https://onlinelibrary.wiley.com/doi/10.1002/spe.3075

[EMMG15] B. Ericson, S. Moore, B. Morrison, M. Guzdial. Usability and Usage of Interactive Features in an Online Ebook for CS Teachers. In *Proceedings of the Workshop in Primary and Secondary Computing Education*. Pp. 111–120. ACM, London United Kingdom, Nov. 2015.
doi:10.1145/2818314.2818335
https://dl.acm.org/doi/10.1145/2818314.2818335

[Eri16] K. A. Ericsson. Summing Up Hours of Any Type of Practice Versus Identifying Optimal Practice Activities: Commentary on Macnamara, Moreau, & Hambrick (2016). *Perspectives on Psychological Science* 11(3):351–354, May 2016.
doi:10.1177/1745691616635600
http://journals.sagepub.com/doi/10.1177/1745691616635600

[HC18] B. Harrington, N. Cheng. Tracing vs. Writing Code: Beyond the Learning Hierarchy. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. Pp. 423–428. ACM, Baltimore Maryland USA, Feb. 2018.
doi:10.1145/3159450.3159530
https://dl.acm.org/doi/10.1145/3159450.3159530

[HCA$^+$16] M. S. Hagger, N. L. D. Chatzisarantis, H. Alberts, C. O. Anggono, C. Batailler, A. R. Birt, R. Brand, M. J. Brandt, G. Brewer, S. Bruyneel, D. P. Calvillo, W. K. Campbell, P. R. Cannon, M. Carlucci, N. P. Carruth, T. Cheung, A. Crowell, D. T. D. De Ridder, S. Dewitte, M. Elson, J. R. Evans, B. A. Fay, B. M. Fennis, A. Finley, Z. Francis, E. Heise, H. Hoemann, M. Inzlicht, S. L. Koole, L. Koppel, F. Kroese, F. Lange, K. Lau, B. P. Lynch, C. Martijn, H. Merckelbach, N. V. Mills, A. Michirev, A. Miyake, A. E. Mosser, M. Muise, D. Muller, M. Muzi, D. Nalis, R. Nurwanti, H. Otgaar, M. C. Philipp, P. Primoceri, K. Rentzsch, L. Ringos, C. Schlinkert, B. J. Schmeichel, S. F. Schoch, M. Schrama, A. Schütz, A. Stamos, G. Tinghög, J. Ullrich, M. vanDellen, S. Wimbarti, W. Wolff, C. Yusainy, O. Zerhouni, M. Zwienenberg. A Multilab Preregistered Replication of the Ego-Depletion Effect. *Perspectives on Psychological Science* 11(4):546–573, July 2016.
doi:10.1177/1745691616652873
http://journals.sagepub.com/doi/10.1177/1745691616652873

[Joh91] L. F. John. Modules: Providing a Flexible User Environment. In *Proceedings of the Fifth Large Installation Systems Administration Conference*. Pp. 141–152. San Diego, CA, Oct. 1991.

[KCB$^+$21] G. M. Kurtzer, Cclerget, M. Bauer, I. Kaneshiro, D. Trudgian, D. Godlove. hpcng/singularity: Singularity 3.7.3. Apr. 2021.
doi:10.5281/ZENODO.1310023
https://zenodo.org/record/1310023

[KCN$^+$20] J. Kumuthini, M. Chimenti, S. Nahnsen, A. Peltzer, R. Meraba, R. McFadyen, G. Wells, D. Taylor, M. Maienschein-Cline, J.-L. Li, J. Thimmapuram, R. Murthy-Karuturi, L. Zass. Ten simple rules for providing effective bioinformatics research

support. *PLOS Computational Biology* 16(3):e1007531, Mar. 2020.
doi:10.1371/journal.pcbi.1007531
https://dx.plos.org/10.1371/journal.pcbi.1007531

[KFMG20] J. Kunkel, W. Filinger, C. Meesters, A. Gerbes. The HPC Certification Forum: Toward a Globally Acknowledged HPC Certification. *Computing in Science & Engineering* 22(4):110–114, July 2020.
doi:10.1109/MCSE.2020.2996073
https://ieeexplore.ieee.org/document/9121599/

[KKJ+15] K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, N. L. Bier. Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*. Pp. 111–120. ACM, Vancouver BC Canada, Mar. 2015.
doi:10.1145/2724660.2724681
https://dl.acm.org/doi/10.1145/2724660.2724681

[LAF+04] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon, L. Thomas. A multi-national study of reading and tracing skills in novice programmers. In *Working group reports from ITiCSE on Innovation and technology in computer science education*. Pp. 119–150. ACM, Leeds United Kingdom, June 2004.
doi:10.1145/1044550.1041673
https://dl.acm.org/doi/10.1145/1044550.1041673

[LDO+13] S. Lampa, M. Dahlö, P. I. Olason, J. Hagberg, O. Spjuth. Lessons learned from implementing a national infrastructure in Sweden for storage and analysis of next-generation sequencing data. *GigaScience* 2(1):9, Dec. 2013.
doi:10.1186/2047-217X-2-9
https://academic.oup.com/gigascience/article-lookup/doi/10.1186/2047-217X-2-9

[LFT09] R. Lister, C. Fidge, D. Teague. Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *ACM SIGCSE Bulletin* 41(3):161–165, Aug. 2009.
doi:10.1145/1595496.1562930
https://dl.acm.org/doi/10.1145/1595496.1562930

[MJL+21] F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, A. Wilm, M. Holtgrewe, S. Rahmann, S. Nahnsen, J. Köster. Sustainable data analysis with Snakemake. *F1000Research* 10:33, Apr. 2021.
doi:10.12688/f1000research.29032.2
https://f1000research.com/articles/10-33/v2

[MNB+17] M. R. Munafò, B. A. Nosek, D. V. M. Bishop, K. S. Button, C. D. Chambers, N. Percie Du Sert, U. Simonsohn, E.-J. Wagenmakers, J. J. Ware, J. P. A. Ioannidis. A

manifesto for reproducible science. *Nature Human Behaviour* 1(1):0021, Jan. 2017.
doi:10.1038/s41562-016-0021
https://www.nature.com/articles/s41562-016-0021

[noa] Jinja 2.
https://jinja.palletsprojects.com/

[OAR+23] E. A. Omer, S. Abdelfatah, M. Riedl, C. Meesters, A. Hildebrandt, T. Efferth. Coronavirus Inhibitors Targeting nsp16. *Molecules* 28(3):988, Jan. 2023.
doi:10.3390/molecules28030988
https://www.mdpi.com/1420-3049/28/3/988

[PAR+22] A. Pohler, S. Abdelfatah, M. Riedl, C. Meesters, A. Hildebrandt, T. Efferth. Potential Coronaviral Inhibitors of the Nucleocapsid Protein Identified In Silico and In Vitro from a Large Natural Product Library. *Pharmaceuticals* 15(9):1046, Aug. 2022.
doi:10.3390/ph15091046
https://www.mdpi.com/1424-8247/15/9/1046

[PH06] D. Parsons, P. Haden. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*. ACE '06, pp. 157–163. Australian Computer Society, Inc., AUS, 2006.

[PTB+18] E. Pérez-Wohlfeil, O. Torreno, L. J. Bellis, P. L. Fernandes, B. Leskosek, O. Trelles. Training bioinformaticians in High Performance Computing. *Heliyon* 4(12):e01057, Dec. 2018.
doi:10.1016/j.heliyon.2018.e01057
https://linkinghub.elsevier.com/retrieve/pii/S2405844018329797

[RKPV14] M. Rubio-Sánchez, P. Kinnunen, C. Pareja-Flores, Velázquez-Iturbide. Student perception and usage of an automated programming assessment tool. *Computers in Human Behavior* 31:453–460, Feb. 2014.
doi:10.1016/j.chb.2013.04.001
https://linkinghub.elsevier.com/retrieve/pii/S0747563213001040

[RPHH18] A. G. S. Raj, J. M. Patel, R. Halverson, E. R. Halverson. Role of Live-coding in Learning Introductory Programming. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*. Pp. 1–8. ACM, Koli Finland, Nov. 2018.
doi:10.1145/3279720.3279725
https://dl.acm.org/doi/10.1145/3279720.3279725

[SHM+22] C. Schmidt, J. Hanne, J. Moore, C. Meesters, E. Ferrando-May, S. Weidtkamp-Peters, members of the NFDI4BIOIMAGE initiative. Research data management for bioimaging: the 2021 NFDI4BIOIMAGE community survey. *F1000Research* 11:638, Sept. 2022.

doi:10.12688/f1000research.121714.2
https://f1000research.com/articles/11-638/v2

[SWCR18] J. Stockard, T. W. Wood, C. Coughlin, C. Rasplica Khoury. The Effectiveness of Direct Instruction Curricula: A Meta-Analysis of a Half Century of Research. *Review of Educational Research* 88(4):479–507, Aug. 2018.
doi:10.3102/0034654317751919
http://journals.sagepub.com/doi/10.3102/0034654317751919

[TAO$^+$23] T. Tang, A. M. Abuhmaid, M. Olaimat, D. M. Oudat, M. Aldhaeebi, E. Bamanger. Efficiency of flipped classroom with online-based teaching under COVID-19. *Interactive Learning Environments* 31(2):1077–1088, Feb. 2023.
doi:10.1080/10494820.2020.1817761
https://www.tandfonline.com/doi/full/10.1080/10494820.2020.1817761

[TGD$^+$18] The Bioconda Team, B. Grüning, R. Dale, A. Sjödin, B. A. Chapman, J. Rowe, C. H. Tomkins-Tinch, R. Valieris, J. Köster. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nature Methods* 15(7):475–476, July 2018.
doi:10.1038/s41592-018-0046-7
https://www.nature.com/articles/s41592-018-0046-7

[UTM$^+$13] I. Utting, A. E. Tew, M. McCracken, L. Thomas, D. Bouvier, R. Frye, J. Paterson, M. Caspersen, Y. B.-D. Kolikant, J. Sorva, T. Wilusz. A fresh look at novice programmers' performance and their teachers' expectations. In *Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports*. Pp. 15–32. ACM, Canterbury England United Kingdom, June 2013.
doi:10.1145/2543882.2543884
https://dl.acm.org/doi/10.1145/2543882.2543884

[XLN$^+$19] B. Xie, D. Loksa, G. L. Nelson, M. J. Davidson, D. Dong, H. Kwik, A. H. Tan, L. Hwa, M. Li, A. J. Ko. A theory of instruction for introductory programming skills. *Computer Science Education* 29(2-3):205–253, July 2019.
doi:10.1080/08993408.2019.1565235
https://www.tandfonline.com/doi/full/10.1080/08993408.2019.1565235

[XRC$^+$17] H. Xu, T. Russell, J. Coposky, A. Rajasekar, R. Moore, A. De Torcy, M. Wan, W. Shroeder, S.-Y. Chen. Integrated Rule-Oriented Data System. In *iRODS Primer 2*. Pp. 3–11. Springer International Publishing, Cham, 2017.
doi:10.1007/978-3-031-02309-5$_2$
https://link.springer.com/10.1007/978-3-031-02309-5_2