Resource Sheet

Enabling Open Science Through Research Code

Episode 2: Enabling Reproducibility through Research Code

Session information

Webpage	https://rsse.africa/events-rsse-africa/2024-11-14/
Date	14 November 2024
Time	08:30 - 10:00 am UTC (your time zone)
Facilitators	Saranjeet Kaur Bhogal Anelda van der Walt Jyoti Bhogal
Co-facilitators	Mireille Grobbelaar
Speakers	• Kozo Nishida, Tokyo University of Agriculture and Technology, Project Researcher • Peter van Heusden, South African National Bioinformatics Institute / University of the Western Cape, Bioinformatician • Radovan Bast, UiT- University of Tromsø, Norway / CodeRefinery, Research Software Engineer
Zoom registration	https://us06web.zoom.us/meeting/register/tZcpcu-ppjstGNJwEt1aKRjR6r2aC7Q4qE9Y

Schedule

Time	Activity	Speakers
8:30 UTC	Giving everyone time to join	**Facilitator: Anelda**
8:35 UTC	Welcome	**Facilitator: Saranjeet**
8:45 UTC	Overview & Recap from Episode 1	**Facilitator: Anelda**
8:50 UTC	Introducing our speakers *Tell us a bit about your background and your current role?* • *What did you study and where?* • *Where do you currently work/study?* • *What is the title of your current role?* • *Give a one-sentence summary of what you do in this role*	**Facilitator: Saranjeet** Kozo Nishida Peter van Heusden Radovan Bast
9:10 UTC	Polls	**Facilitator: Anelda**
9:15 UTC	Panel discussion • *What is reproducibility, and why do you think it is important? Can you mention examples of times when you encountered analyses that were not reproducible or tools that didn't enable reproducibility?* • *What steps do you take to ensure your code enables reproducibility in your research project, especially when developing code that supports a research article?* • *It's not always easy to start from scratch. Why and how can*	**Facilitator: Anelda/Jyoti** Kozo Nishida Peter van Heusden Radovan Bast

Time	Activity	Speakers
	you embark on a journey of reproducibility using other researchers' code? • *Do you have any suggestions for easily implementable, small changes that can make a difference (such as having a good directory structure)?* • *Do you use containers to manage versions and dependencies? Yes/No and why? What alternatives are there?* • *There are numerous learning resources to help researchers adopt tools and practices that can enable reproducibility.* *Would you like to point our audience to anything specific? Can they contribute to these resources?*	
9:55 UTC	Wrap up and next steps	**Facilitator: Anelda**

Resources and info from partners

RSSE Africa

- Website: https://rsse.africa
- Sign up for our newsletter: https://talarify.us14.list-manage.com/subscribe?u=35d5db26d3b108b9ef9b9ac43&id=55e9f5a692
- Join our LinkedIn group, where you can also share information with the broader community: https://www.linkedin.com/groups/12903402/

RSE Asia

- Website: https://rse-asia.github.io/RSE_Asia/
- For the latest news, events, activities, and opportunities, follow us on our LinkedIn page (https://www.linkedin.com/company/rse-asia-association/)
- To join the RSE Asia community, please fill out our short Community Membership Form (https://docs.google.com/forms/d/1XSxDaTJzcNyGeDYXyJNVg1TDCo7un18PLFNiK6_jL2g/edit)

AREN

- Website: https://africanrn.org/
- Sign up: https://docs.google.com/forms/d/e/1FAIpQLSeeFkD5A4D9l6ncQWjKBiI-GqBOzL-JMe7Fx3ijUYEjHjDUoQ/viewform

CodeRefinery

- Website: https://coderefinery.org/
- Lesson portfolio (all CC-BY and contributions/ideas welcome): https://coderefinery.org/lessons/
- Chat where the project and community discusses (everybody can join; 485 subscribers): https://coderefinery.zulipchat.com/
- Next big online workshop: March or April 2025 (we have just started planning it)
- Many CodeRefinery folks are part of Nordic RSE: https://nordic-rse.org/

ReSA

- Website: https://www.researchsoft.org/
- Sign up for the newsletter: https://www.researchsoft.org/news/
- The Amsterdam Declaration on Funding Research Software Sustainability
    - Become a signatory: https://adore.software/sign/

Questions for the audience

- What discipline do you work in?
    - (Kozo) Bioinformatics. I work in software development for data integration and data analysis in bioinformatics. The main focus is metabolomics, but I also work with other omics datasets. Regardless of the datasets, my goal is automation and reproducible analysis. To be honest, I'm not interested in scientific discovery. I'm more interested in improving the efficiency of research.
- How much of your working time do you spend coding?
    - (Kozo) A few days per month. Honestly, I spend more time writing Slack messages and emails than coding. The main reason is the number of organizations and collaborators I'm involved with. Interacting with many people is helpful for gaining knowledge, but I think it's a problem that I haven't been able to secure enough time for coding. Unfortunately, I haven't found the best practice for increasing my coding time yet.
- Can you share your favourite tip for enabling reproducibility - it doesn't necessarily have to be related to coding?
    - (Kozo) My tip is creating a package in Python or R and making it public. I think publishing Jupyter Notebook or RMarkdown also helps improve reproducibility, but packaging the functions used in those notebooks makes them even more reproducible. And for bioinformatics R packages, I think it is better to create a Bioconductor package rather than a CRAN package. Bioconductor provides a different package build system, version management, and Docker environment compared to CRAN. I believe Bioconductor offer higher reproducibility. I think the best resource for Bioconductor package building is https://contributions.bioconductor.org. Bioconductor is also developing a Carpentries lesson for building

Bioconductor package, but it is still in the pre-alpha stage.

Questions for organisers, facilitators and/or speakers

- How much time do you spend on reorganizing the code itself? Especially to make it reproducible and general enough for future projects.
    - It takes a lot (months) to move from markdown to package.
    - Even longer if you move to another position
- What about code snippets you find on Stack Overflow or other online tutorials etc? What about code that is suggested by ChatGPT and similar tools? Can we reuse? Can we share? Is it clear what we are allowed to do?
    - https://stackoverflow.blog/2009/06/25/attribution-required/
    - https://botpress.com/blog/are-there-any-legal-or-copyright-concerns-when-using-chatgpt-generated-content#:~:text=According%20to%20OpenAI's%20Content%20Policy,paid%20plan%2C%20or%20their%20API.
- How can we apply reproducibility best practices to qualitative projects?
    - https://openworking.wordpress.com/2019/02/11/what-does-reproducibility-mean-for-qualitative-research/#:~:text=Limits%20of%20Reproducibility&text=Sebastian%20argued%20that%20reproducibility%20and,are%20quite%20impossible%20to%20recreate.
    - https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-019-0707-y
    - https://scientificallysound.org/2019/09/24/reproducible-research-practices-in-qualitative-research-part-1/

Shared resources from the session

- Suggestions from the audience for enhancing reproducibility:
    - GitHub for documentation and Jupyter files for code
    - Pipelines netflow, GitHub, versioning
    - Using virtual environments (renv for R, venv for Python)
    - Documentation, documentation and documentation.
    - annotation - nothing worse than trying to work out why you did something a few weeks later

    - if working with python, pip freeze is helpful to get a full list of what packages you might have installed
    - document everything - and try and avoid "clever" one liners in the shell that won't end up in your documentation
    - Building markdown documents with Quarto and packaging code as R package
    - One thing that really helps me is closing and reopening the IDEs that I use frequently - so that I know sooner if something is breaking
    - I like Jupyter notebooks but you have to be disciplined in how you use them (i.e. keep it strictly top-to-bottom, add Markdown cells to the document)
    - Teamwork is important - https://academic.oup.com/bioinformatics/article/40/11/btae632/7831429
    - JOSS review checklist: https://joss.readthedocs.io/en/latest/review_checklist.html
    - https://reprohack.github.io/reprohack-hq/
    - https://dvc.org/ - An open-source tool designed to facilitate the management and versioning of data, models, and experiments in data science and machine learning projects.
    - Another aspect that I have found apart from reproducing code, is reproducing the data itself for analysis especially when it comes to very sensitive datasets by creating synthetic data for sharing with public users, they are some resources that can help with this such as synthpop: https://thomvolker.github.io/osf_synthetic/osf_synthetic_workshop.html
    - https://git-lfs.com
    - https://www.w3schools.com/python/ref_keyword_assert.asp
    - https://en.wikipedia.org/wiki/Pair_programming
- Mastering Docker: A Guide to Containerizing Tools with Incomplete Documentation - link
- Non-open or not-clearly-open licenses make collaborative development and reproducibility really hard. Avoid creating custom licenses and prefer standard licenses where compatibility is clear
- Why should you care about reproducible code — and how to get started? Link
- Tools for versioning (larger) data sets: https://dvc.org/, https://git-annex.branchable.com/, and https://git-lfs.com (proprietary).

- An open-source tool designed to facilitate the management and versioning of data, models, and experiments in data science and machine learning projects.
- https://www.docker.com/resources/what-container /
- my longest-surviving piece of code: https://github.com/pvanheus/sendmail-turing - written in 1998 and now still runnable because it is available in a container!
- https://bioconda.github.io/
- https://www.nextflow.io/
- https://snakemake.readthedocs.io/en/stable/
- https://docs.sylabs.io/guides/3.5/user-guide/introduction.html
- bulker: https://bulker.databio.org/ <- use if you want
- While working on a large scale, say 100 or 1000 files, you must use workflow: https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html
- https://homepage.hackmd.io/
- https://streamlit.io/
- Keep all your dependencies together:
    - https://docs.python.org/3/library/venv.html
    - https://rstudio.github.io/renv/articles/renv.html
- https://github.com/astral-sh/uv
- https://martinfowler.com/bliki/FrequencyReducesDifficulty.html
- All the dependencies should be mentioned inside field, say readme or something...
- Spack (https://spack.io/) is an excellent tool when it offers the applications you need. EasyBuild (https://docs.easybuild.io/) also seems like a valuable resource, although I haven't had the chance to try it yet.
    - Lots of HPC people like Spack because it saves the details of how software is compiled and this means that you can optimise this for your computing environment and then save and share those configs.
- https://coderefinery.github.io/reproducible-research/
- adopt tools and practices that can enable reproducibility.
- https://carpentries.org
- https://training.galaxyproject.org/
- https://usegalaxy.org/
- https://ubinfie.github.io/2024/08/16/adding-to-bioconda-quickguide.html
- CodeRefinery lesson portfolio: https://coderefinery.org/lessons/
- https://carpentries.org/become-instructor/

- https://book.the-turing-way.org/index.html

Upcoming events from around the world

- **Curious about containers?** Learn how to create reproducible R environments with containers! Join **Noam Ross**, disease ecologist & rOpenSci Executive Director, as he dives into the Rocker Project & more! Registration: https://r-consortium.org/webinars/containerization-and-r-for-reproducibility.html
- **Coderefinery**: The next online workshop starts in April. They have a concept to bring your own class (bring your own students to learn from the workshop)
- Registration is now open for the second round of **Research Software Practices in the Social Sciences** workshops: https://www.software.ac.uk/news/registration-now-open-second-round-research-software-practices-social-sciences-workshops
- Many more opportunities are listed at https://rsse.africa/events