Research software engineers accelerate translational health research

Authors

David Horsfall[1,2]*, Jonah Cool[3], Simon Hettrick[4], Angela Oliveira Pisco[5], Neil Chue Hong[6], Muzlifah Haniffa[2,1]*

Affiliations

[1] Biosciences Institute, Newcastle University, Newcastle upon Tyne, NE2 4HH, UK
[2] Wellcome Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SA, UK
[3] Chan Zuckerberg Initiative, Redwood City, California, USA
[4] University of Southampton and the Software Sustainability Institute, Southampton, SO17 1BJ, UK
[5] Chan Zuckerberg Biohub, San Francisco, California, USA (Present address: Insitro, Inc., South San Francisco, CA)
[6] University of Edinburgh and the Software Sustainability Institute, Edinburgh, EH8 9YL, UK

* corresponding authors

Research software engineering is central to data-driven biomedical research, but the role is often undervalued and poorly understood.

Biomedical and clinical research have become increasingly data-driven. Transforming large amounts of data into new discoveries requires cutting-edge analytical approaches, as well as new infrastructure to provide a foundation upon which algorithmic advances can build. Greater collaboration with outside fields such as software engineering and computer science has driven new advances in computational biology — with experts in these domains working alongside biomedical researchers and clinicians to acquire cross-domain expertise.

Published papers are increasingly dependent on algorithms and software that underpin the reported research. With the increasingly foundational role of computational approaches in biomedical science comes challenges associated with reproducibility of results and robustness of underlying code. A 2016 survey of 1,500 scientists found that over 70% had tried and failed to reproduce another scientist's experiments[1]. The same year, the FAIR Guiding Principles were published, aimed at enhancing the reusability of scientific data. Transparency of software code is a prerequisite for reproducibility and is necessary to understand the provenance of research data and insights. To improve the transparency of methods and interoperability of data there is a rapidly growing need for well-engineered solutions that transcend a single lab and can be used by a large number of scientists.

Research software engineering is an emerging field focused on addressing these core challenges through a unique skill set that enhances the value and usage of scientific data. Research software engineering can facilitate interdisciplinary science and accelerate translational research through efficient data management and equitable data provision.

An emerging role

Research software engineering combines professional software engineering expertise with an intimate understanding of research. The focus is to deliver best practices through the application of foundational software engineering practices such as version control, testing and automation, while at the same time ensuring the data output remains scientifically valid and accurate. The Research Software Engineer (RSE) speaks the language of professional engineering and understands fundamental research methods. From this unique position, RSEs can think differently about research questions and spur innovative solutions which scientists and data analysts alone might not reach.

The application of professional software engineering is critical for scaling and reproducibility of scientific output, especially as researchers grapple with the sheer size and volume of datasets, as well as an abundance of different analysis methods. The impact of research software across science is huge; consider, for example, the industrial scaling of centralised genome browser resources such as Ensembl that revolutionised the biosciences with massive infrastructure and engineering projects.

The concept of the RSE has only existed for a decade and has grown rapidly, establishing the importance of the discipline across various scientific domains[2]. Since the idea was first proposed at an event at The Queen's College, Oxford in March 2012[3], the movement has spread to a substantial international community, with 10 established associations in the United Kingdom, mainland Europe, Africa, Asia, Australia, and North America. In the United Kingdom, at least 38 universities have their own centralised RSE groups that researchers can use to access skilled software professionals to develop the software tools they need for their research.

Through support from organisations like the Software Sustainability Institute[4], the RSE community has helped develop several initiatives that champion open science and reproducibility in the life sciences. Any researcher who writes code, such as bioinformaticians, can align with RSE communities and benefit from exposure, training and peer support. Other resources like The Turing Way[5] provide a handbook for reproducible, ethical and collaborative data science. As awareness improves, RSEs are being increasingly embedded within research teams, and this in turn increases accountability and enhances trust in the scientific results delivered by software. Ways to engage, receive training and work effectively with RSEs are in Box 1.

Data-driven science

Emerging technologies and big data open up exciting new opportunities for scientific discovery. Artificial intelligence has the potential to extract new actionable insights from the complexity of human health and disease, with prospective applications in biomedicine including image-based diagnostics and the discovery of new, more effective treatments. Emerging computational approaches with the potential to transform biomedicine must be underpinned by robust and scalable software, ideally from professionals who sit between research and technology, as exemplified by AlphaFold — an AI system developed by

DeepMind and EMBL-EBI to provide open access to over 200 million protein structure predictions[6].

While research software engineering can play a crucial part in the research lifecycle, the recognition of its importance does not yet match that of data generation and analysis. That said, RSEs are a key driver for research success, dissemination and impact. By investing in the adoption of FAIR principles throughout the data pipeline and extending those principles to software[7], RSEs can transform research data output from being seen as a final resting place into a dynamic, collaborative resource in an active ecosystem of tools and infrastructure.

Team science and translational research

The research landscape is seeing increasingly large interdisciplinary collaborations across institutions, which often generate high-impact research[8]. Approaches that integrate biological and clinical knowledge lead to innovations for improving health outcomes. Modern science relies on many people and many different skills to conduct research from community managers to people who produce training materials.

One example of global collaborative science is the Human Cell Atlas initiative, which aims to characterise and map every cell type in the human body. This international consortium has over two thousand members in over eighty countries and invests in building capacity through multidisciplinary teams that champion open science, including software engineers focused on data storage, sharing, browsing and dissemination. Their data and findings are shared openly with the broader scientific community, which accelerates discoveries and deepens collaboration among researchers around the world. RSEs played a fundamental role in the rapid coordination and deployment of the consortium's centralised COVID-19 Data Portal[9].

While many funders support software development, less money has historically been available for the critical work of software maintenance. Fortunately, a growing number of funders are seeking to address this problem. Schmidt Futures recently announced the creation of the $40 million Virtual Institute of Scientific Software to fund the maintenance of researcher-written code[10]. The Chan Zuckerberg Initiative, a philanthropic organisation that is dedicated to building the future of science by funding efforts like the Human Cell Atlas, has also pledged $40 million through its Essential Open Source Software for Science program. This provides support for ongoing maintenance of widely used open source scientific software that is critical to maintain the ecosystem, which is often overlooked by discovery science funding mechanisms.

RSEs drive clinical translation of research findings. By delivering data through web applications, for example, RSEs remove the technical burden from clinicians, students, investigators and industry partners. Since the only requirement is internet access and a web browser, this significantly improves global, equitable access to research data. Similarly, if data visualisation and analysis tools are more readily available through intuitive point-and-click interfaces, research teams around the world can collaborate more easily. The development of open source scientific tools and resources for single-cell biology data, such as the Chan Zuckerberg CELL by GENE platform[11], the Human Developmental Cell Atlas[12]

and the Cambridge Cell Atlas[13], helps scientists explore and visualise high-dimensional single-cell datasets to derive scientific insights. Tools and resources like these empower researchers to access data when it suits them, and facilitate collaborative research to improve data generation, analysis, biological interpretation and the clinical application of research findings. Essential design considerations for the success and development of these web applications are in Box 2.

While the scientific landscape has changed and new roles and expertise have become increasingly important, the judging of research excellence has not kept pace. Assessment of research success tends to focus on individual achievements, such as a published article or a successful grant, instead of cumulative progress that can lead to breakthroughs. Mechanisms that reward individual researchers inevitably undervalue those in roles essential to collective research projects, such as RSEs, lab managers and research technicians. That said, there are new awards and accolades that are focused on recognizing the contributions of all roles within research. The Hidden REF, for example, celebrates all research outputs and recognises everyone who contributes to their creation[14]. However, these initiatives are frequently organised by volunteers, so their impact is limited until they can attract greater awareness and funding.

There is a need for greater acknowledgement and support for emerging roles such as research software engineering from all stakeholders, including funders, research organisations, learned societies and researchers with traditional scientific backgrounds. An important first step is the regular citation and acknowledgement of software and the contributions made by research software engineering to scientific papers[15]. Many researchers do not know that software is citable; frameworks such as the CZ Software Mentions Dataset[16] elevate software to a research output. Proper credit for software tools and their utility is key to ensuring the role of RSEs is fully understood and recognized by the broader scientific community. If labs cite software they use in publications, and encourage training and peer support for software development, they might then build the skills and confidence to publish their own software. This supports collaboration and allows others to cite the software, leading to a cultural cycle of valuing software in research (Fig. 1).

A deluge of data

A critically undervalued part of research software engineering is the creation of new data models, data infrastructure, and file standards for emerging technologies. This work is essential, because it provides the underlying framework for labs to create, store, share and collaborate on research data at scale. Technical frameworks for new innovative projects haven’t yet been created, and only trained engineers with a solid understanding of research can deliver products at the required robustness and scale. Individual labs are infrequently motivated to take this work on and yet a rapidly growing cross-section of science benefits from, and indeed is reliant upon, their efforts. For these projects, RSEs need to understand the structure of the data being generated, assess how the data will be consumed and anticipate future challenges and innovations.

The Open Problems in Single Cell Analysis[17] project provides an open source, community-driven platform for continuously updated benchmarking of formalised tasks in

single-cell analysis. Algorithms for tasks such as batch integration, or comparison of data denoising methods, can now be easily benchmarked using the platform. By driving community convergence on new standards, data can be stored with integrity, ported between labs, and easily interrogated. Recently, the Open Microscopy Environment consortium that has maintained a common data model for bioimaging for the past twenty years described their efforts to create a next-generation file format for bioimaging[18], driven by the need to share large imaging data in the cloud. The adoption of this format was only achieved through significant efforts by RSEs to update existing tools, but critically also required coordination efforts in the community, organising events and gently building consensus.

Software engineering is a discipline rooted in identifying major challenges and then constructing solutions to them, for the benefit of many. In the case of modern biomedicine, the importance of this mindset and skill is growing rapidly. The deluge of data, potential of advanced computational approaches, and increasing impact of team science together create a research environment with RSEs as a critical central component. It is increasingly clear that sharing data openly at the scale it's being generated is not reaching its full potential. Promoting data utility requires not just storage solutions that scale, but performant software and infrastructure solutions by which data can be made interoperable, visualised, and leveraged by experts and non-experts alike. These important contributions need to be recognised and rewarded as biomedical science advances. Research software engineering is poised to revolutionise how the scientific community can democratise not just the data, but the technical infrastructure and mechanism for interacting with it, providing an opportunity to modernise how scientists and the public engage with research narratives.

References

1. Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature Publishing Group UK* http://dx.doi.org/10.1038/533452a (2016) doi:10.1038/533452a.
2. Woolston, C. Why science needs more research software engineers. *Nature Publishing Group UK* http://dx.doi.org/10.1038/d41586-022-01516-2 (2022) doi:10.1038/d41586-022-01516-2.
3. Hettrick, S. A not-so-brief history of Research Software Engineers. https://www.software.ac.uk/blog/2016-08-17-not-so-brief-history-research-software-engineers-0.
4. The Software Sustainability Institute. https://www.software.ac.uk/.
5. The Turing Way Community *et al.* The Turing Way: A Handbook for Reproducible Data Science. (2019) doi:10.5281/zenodo.3233986.

6. AlphaFold Protein Structure Database. https://alphafold.ebi.ac.uk/.
7. Chue Hong, N. P. *et al.* FAIR Principles for Research Software (FAIR4RS Principles). (2022) doi:10.15497/RDA00068.
8. Wuchty, S., Jones, B. F. & Uzzi, B. The increasing dominance of teams in production of knowledge. *Science* **316**, 1036–1039 (2007).
9. COVID-19 Cell Atlas. https://www.covid19cellatlas.org.
10. Matthews, D. Ex-Google chief's venture aims to save neglected science software. *Nature* **607**, 410–411 (2022).
11. GitHub - CZ CELLxGENE platform. https://github.com/chanzuckerberg/cellxgene.
12. The Human Developmental Cell Atlas. https://developmental.cellatlas.io/.
13. The Cambridge Portal of the Human Cell Atlas. https://www.cambridgecellatlas.org/.
14. Derrick, G. & Hettrick, S. Time to celebrate science's 'hidden' contributors. *Nature Publishing Group UK* http://dx.doi.org/10.1038/d41586-022-00454-3 (2022) doi:10.1038/d41586-022-00454-3.
15. Katz, D. S. *et al.* Recognizing the value of software: a software citation guide. *F1000Res.* **9**, 1257 (2020).
16. Hutson, M. Hunting for the best bioscience software tool? Check this database. *Nature* (2023) doi:10.1038/d41586-023-00053-w.
17. Burkhardt, D. B. Open Problems in Single Cell Analysis. *Open Problems in Single Cell Analysis* https://openproblems.bio/.
18. Moore, J. *et al.* OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. *Nat. Methods* **18**, 1496–1498 (2021).

Acknowledgements

We are grateful to Toby Hodges from The Carpentries for training recommendations, Chloe Admane for figure work, and Aidan Maartens for critical reading of the manuscript.

Competing interests

The authors declare no competing interests.

Funding

S.H. and N.C.H. time was partly supported by the UK Research Councils through grant EP/H043160/1. M.H. is funded by a Wellcome Senior Research Fellowship in Clinical Sciences (223092/Z/21/Z) and Wellcome core grant (220540/Z/20/A).

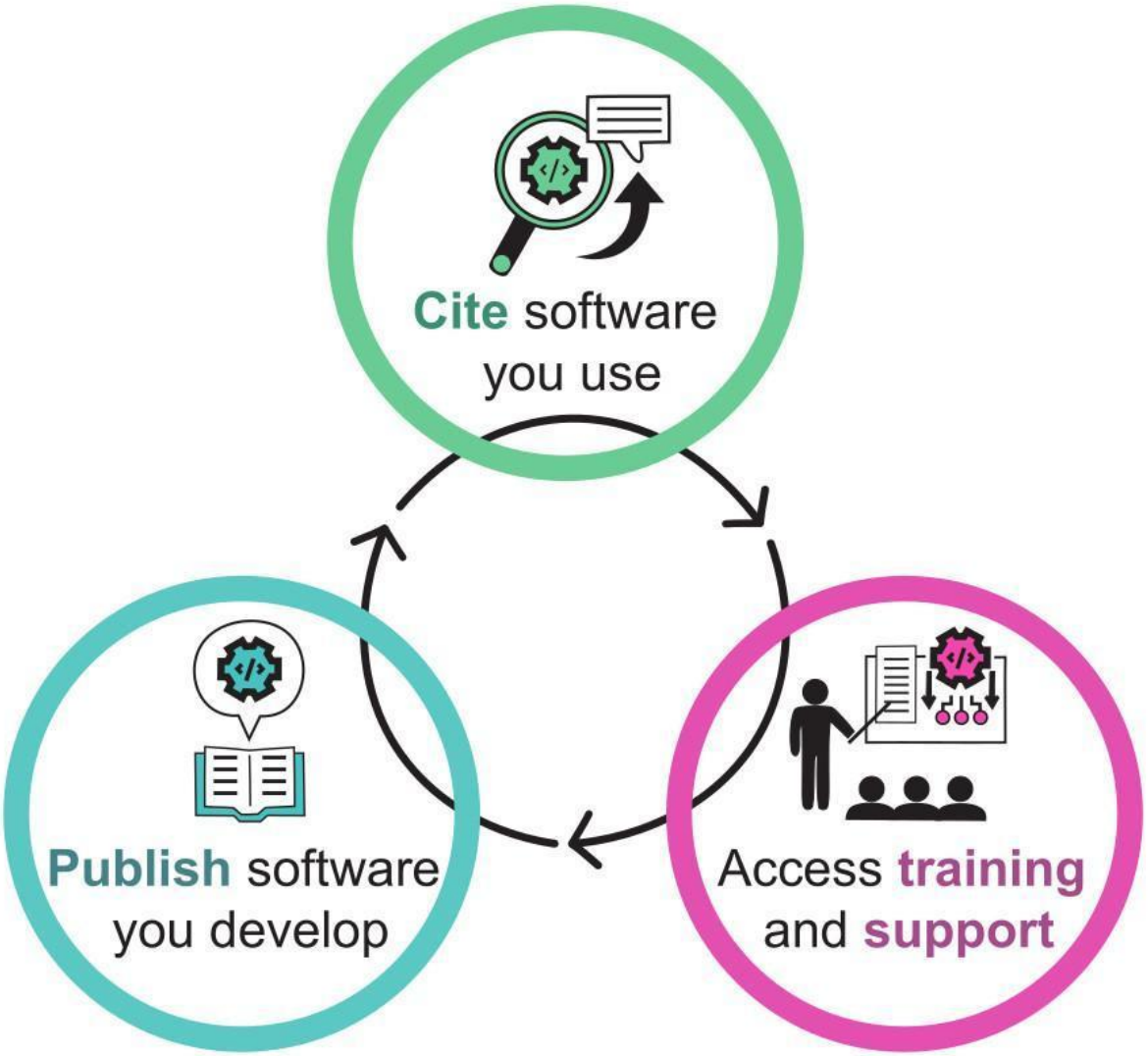
Author Contributions

D.H., J.C., M.H.: Conceptualization, Writing – Original Draft, Writing – Review & Editing

S.J.H., A.O.P., N.C.H.: Conceptualization, Writing – Review & Editing

Figure 1 | A community cycle of valuing software in the research process

Labs should cite software they use in publications, encourage training and peer support, and eventually have the skills and confidence to publish their own software.



Box 1 | How researchers can engage and work effectively with research software engineers

Get in touch with your local RSE group (society-rse.org/international-rse-organisations) or contact the Society of Research Software Engineering (society-rse.org).
Training in foundational software development and data science is strongly encouraged for everyone working with research software and data. Training can be delivered through The Carpentries (carpentries.org), a leading community that builds global capacity in essential software skills for conducting efficient, open, and reproducible research. They offer workshops on version control, Unix shell, Python, R and several others focused on data literacy.
Help **raise awareness of the software used in research** by citing it in publications. Giving recognition supports proper credit of software tools and their utility, which is key to ensuring the role of RSEs is fully understood and recognized by the broader scientific community.
Publish your own software to enable peer-review, validation, and reproducibility of findings. Publishing software supports collaboration and reuse, and encourages the building on the work of others.
Any researcher who writes code, such as bioinformaticians, can benefit from engaging with RSE communities. **Peer support** can introduce ideas and approaches that others have found helpful, offer a sense of belonging and build self-esteem and confidence. ReproHacks (reprohack.org) are events where participants aim to reproduce scientific results detailed in published papers. Open Life Science (openlifesci.org) is a mentoring and training program for Open Science ambassadors. CodeCheck (codecheck.org.uk) is a framework for independent execution of computations underlying scholarly research articles. The Turing Way (the-turing-way.netlify.app) is a handbook to reproducible, ethical and collaborative data science.
Security and privacy considerations when working with data are important, and sometimes regulated. Research projects with embedded RSE skills can leverage technical expertise to inform wider decision making.

Box 2 | Design considerations for accelerating translational research

Remove technical burden of accessing research data
Unlock data with intuitive point-and-click interfaces
Make data available at any time, from any location
Drive community convergence of new scalable data models and file standards
Modernise the method of engagement with research narrative