

Original Research Articles

Toward the future of scientific publishing through reproducible research artefacts enabled by Neurodesk

Michèle Masson-Trottier, PhD¹^a, Thuy T. Dao, BEng¹, Aswin Narayanan, BEng^{2,3}, Steffen Bollmann, PhD^{1,4}

¹ School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane, Australia, ² Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, Brisbane, Australia, ³ Australian National Imaging Facility, Queensland, Australia, ⁴ Queensland Digital Health Centre, The University of Queensland, Brisbane, Australia

Keywords: open science infrastructure, transparent publishing, software citation, peer review

<https://doi.org/10.52294/001c.143700>

Aperture Neuro

Vol. 5, Issue SI 3, 2025

Computational tools are essential to neuroimaging research, yet their development, use, and citation are inconsistently addressed in current scientific publishing frameworks. Despite increasing adoption of the FAIR and FAIR4RS principles, challenges remain in ensuring the reproducibility, accessibility, and attribution of research software. This article presents how Neurodesk, an open-source, community-driven platform improves the transparency, portability, and sustainability of neuroimaging workflows. Neurodesk uses containerisation to encapsulate complete software environments that are portable across operating systems and computing infrastructures. Each container and workflow can be versioned and assigned a persistent DOI to support formal citation and long-term access. Through concrete use cases, we illustrate how Neurodesk facilitates code review, supports methodological extension, and fosters sustainable tool development. We argue that platforms like Neurodesk provide a practical infrastructure for enacting cultural and technical shifts in publishing, making research software a primary research output. By integrating attribution, standardisation, and reproducibility into everyday research workflows, Neurodesk exemplifies how future scientific publishing can more effectively recognise and preserve computational contributions.

INTRODUCTION

Neuroimaging research draws investigators from diverse disciplinary backgrounds, ranging from neuroscience, psychology and medicine to physics, statistics and computer science. This diversity creates a significant challenge: while neuroimaging analysis demands sophisticated computational techniques and advanced statistical methods, many researchers lack formal training in these areas. This expertise gap has direct consequences for scientific publishing and reproducibility as not all researchers using neuroimaging possess in-depth knowledge of these fields' best practices. Current neuroimaging research faces issues with reviewing and publishing software and code, leading to transparency problems, challenges in long-term accessibility, and bias in peer reviews. This impacts reproducibility and diminishes research quality.

Understanding that reproducibility hinges on access to well-structured data, nearly ten years ago, the research

community adopted the FAIR data principles (Findable, Accessible, Interoperable, and Reusable) as a framework to reduce research waste and improve data stewardship.¹ More recently, the FAIR for Research Software (FAIR4RS) initiative emerged to adapt these principles to research software and code, advocating for improved metadata, versioning, licensing, and preservation strategies.² These efforts reflect a growing recognition that software and code, like data, must be treated as primary research outputs. However, for these efforts to reach their full potential, there needs to be a shift in how scientific publications address these aspects.

One of the most significant challenges in neuroscience and computational (biomedical) research more broadly is the lack of standardisation in code-sharing. This creates barriers to replicating findings within and across research groups and reusing and extending analytical methods. Despite growing awareness, studies show that fewer than 50% of biomedical papers share analytical code, and an even smaller number do so in a structured, reproducible way.^{3,4} Calls for better standardisation of biomedical research soft-

^a Corresponding Author:

Michèle Masson-Trottier, PhD (m.massontrottier@uq.edu.au)

School of Electrical Engineering and Computer Science, The University of Queensland
Brisbane, Australia

ware are not new,⁵ yet implementing open science policies remains inconsistent. Even as more journals support code sharing alongside the already adopted data sharing, guidelines for doing so are often vague, and enforcement of best practices is minimal.³

Numerous factors contribute to researchers' hesitancy - or inability - to share code. These include a lack of training in software engineering practices, time constraints for cleaning and documenting code, and concerns about exposing potential errors or revealing proprietary methods.⁶ For researchers without a formal programming background, there is often a perception that their code "isn't good enough" to be made public.⁷ Unfortunately, this perspective encourages poor reproducibility practices. Also, when fewer people share code, there are fewer examples to learn from, reducing opportunities for collaboration and diminishing the motivation to improve practices in the field. As a community, shifting the norm from "only share perfect code" to "share what you have, with context" is critical, acknowledging that sharing imperfect or exploratory code is a key step toward transparency and collective progress.

Journals such as Aperture, ReScience C, Journal of Open Source Software (JOSS), Magnetic Resonance in Medicine (MRM) and Nature Scientific Data, to name a few, encourage authors to make code available when submitting publications. However, there is little incentive for peer reviewers to thoroughly examine and evaluate code, limiting the effectiveness of these policies. To address this, some journals now offer formal code reviews as part of the publication process to encourage authors to submit their code and boost transparency in science. Still, when researchers make their code available and request a review, it is frequently poorly documented and may be missing dependencies, making the reviewing process time-consuming and difficult.

This situation underscores the need for more straightforward guidelines and better technical support for systemic and cultural shifts to ensure that code sharing is encouraged and done practically and effectively. Disseminating standardised practices for code documentation and dependency management, alongside infrastructure that simplifies the sharing and execution of code, could significantly improve the quality of research outputs and reduce the burden on authors and reviewers.

Taking this one step further, journals and preprint platforms encouraging code-sharing and alternative publications have emerged, promoting executable and reproducible research. For example, NeuroLibre enables the publication of interactive papers with live code execution using Jupyter notebooks.⁸ ReScience C is an open-access, peer-reviewed journal that publishes replicated computational studies with a strong emphasis on code transparency and version control⁹ and [BrainLife.io](https://brainlife.io)¹⁰ provides an integrated platform for executing, sharing, and reproducing neuroimaging workflows via containerised applications and open datasets while also supporting code sharing attribution. These new platforms aim to support cultural and technical shifts while promoting more sustainable research practices.

In parallel with these formal publishing efforts, analysis code and research software are often developed and shared with collaborators or the broader community well before any peer-reviewed publication. In these early stages, tools may circulate through repositories, lab websites, or conference presentations without being captured in traditional academic outputs. As a result, contributions can remain difficult to discover, cite, or credit appropriately, limiting both visibility and recognition for the researchers involved. This gap can hinder collaboration, transparency, and the accumulation of shared infrastructure in fields like neuroimaging. To address this, there is a growing need for dedicated repositories and platforms that support the structured dissemination, citation, and long-term accessibility of research software alongside more conventional scholarly outputs.

Ultimately, long-term accessibility and sustainability of research code remain persistent concerns. Even when authors carefully document the dependencies, operating systems are eventually updated, and code may become obsolete or unusable. This raises serious challenges for reproducibility: researchers may be unable to rerun analyses or verify findings if the original computational environment is no longer accessible. Platforms that preserve both the code and the full execution environment, such as those using containerisation, become essential components of a sustainable, reproducible research ecosystem.

In summary, while computational methods are central to neuroimaging research, the tools and code that enable this work are often undervalued, inconsistently shared, and poorly preserved. Although initiatives like FAIR4RS and new publishing platforms are shifting norms toward greater transparency and sustainability, significant technical and structural barriers remain. Addressing these challenges requires not only improved infrastructure and clearer guidelines, but also a cultural shift in how research software is recognised, cited, and rewarded within the academic system.

In response to these challenges, we developed Neurodesk,¹¹ an open-source initiative that addresses the infrastructural and cultural challenges described above by offering a unified neuroimaging software deployment and reproducibility framework. This article outlines how Neurodesk supports transparent, collaborative, and sustainable research practices through containerised tools and standardised environments across contexts. We describe its technical implementation and how it promotes fair attribution of computational tools, and sets a new standard for transparent, reproducible publishing.

METHODS

THE NEURODESK APPROACH

Neurodesk is an open-source, community-supported platform that facilitates reproducible and accessible neuroimaging analysis across diverse computing environments. While several container-based solutions have been developed to improve reproducibility in neuroimaging (e.g.,

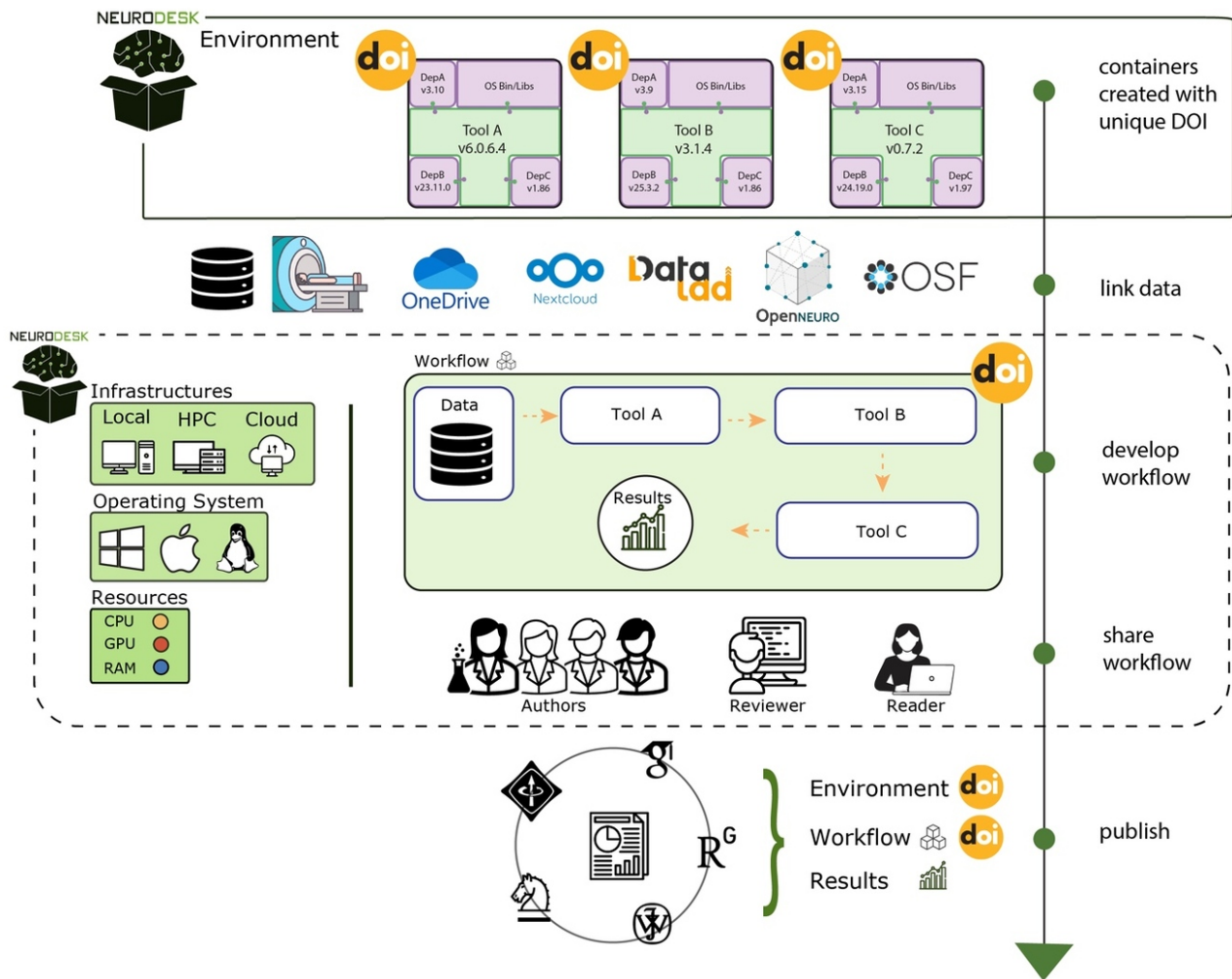


Figure 1. Overview of Neurodesk's infrastructure for reproducible, transparent, and citable neuroimaging workflows. Within Neurodesk, each tool is packaged in a Neurocontainer, a containerised environment that includes all dependencies (identified as DepA, DepB and DepC) and runtime configurations, and assigned a persistent Digital Object Identifier (DOI) to support formal citation. These Neurocontainers, illustrated in the top row, define stable software environments that ensure long-term reproducibility. Once the environment is defined, Neurodesk supports integration with widely used FAIR data-sharing platforms (e.g., Datalad, OpenNeuro), enabling users to build modular workflows from standardised tools and various datasets. These workflows can be executed across various systems, including personal computers, workstations, high-performance computers (HPC), and cloud systems, as well as different operating systems (Windows, macOS, and GNU/Linux). This enables code-sharing, portability across platforms and reproducibility over time. Workflows can also be saved and shared as complete, citable research objects with their own DOI which promotes attribution of both individual tools and complete analytical workflows.

BIDS Apps,¹² BrainLife.io,¹⁰ NeuroLibre⁸), Neurodesk distinguishes itself by supporting consistent execution across local machines, high-performance computing (HPC) systems, and cloud platforms on Windows, macOS, and GNU/Linux. Neurodesk is available both as a desktop application with a graphical user interface (GUI) for intuitive use and through Neurocommand, a lightweight command-line interface that enables scripted deployment and management of tools.

Neurodesk enables researchers to work in standardised environments without concern for underlying system configurations by abstracting software installation and dependency management through containerisation. This modular

system supports a broad ecosystem of widely used neuroimaging tools (e.g., FSL,¹³⁻¹⁵ ANTs,¹⁶ FreeSurfer (freely available for download online <http://surfer.nmr.mgh.harvard.edu/>), MRtrix¹⁷). Moreover, it is designed to run seamlessly across Windows, macOS, GNU/Linux, and remote computing resources. This enables diverse research workflows, from exploratory desktop-based analyses to scalable pipeline execution on HPC systems, while supporting transparency, reproducibility, and long-term software sustainability.

ENHANCING CODE REPRODUCIBILITY THROUGH CONTAINERISATION

To address challenges associated with reproducibility in computational neuroscience, Neurodesk provides a framework to build and deliver neuroimaging software within containerised environments. These containers, called Neurocontainers ([Figure 1](#)), are built using the Neurodocker¹⁸ recipe generator provided by the ReproNim project.¹⁹ Each Neurocontainer includes all dependencies required to execute specific tools, ensuring consistency across systems and over time. This strategy allows researchers to bypass the process of installing individual software natively, and avoid issues caused by dependency and system incompatibilities, thereby enabling analyses to be reproduced on a wide range of systems without modifications to the codebase or environment. To ensure compatibility across diverse computing environments, including those where Docker is restricted for security reasons, such as many university-provided HPCs, Neurocontainers are designed to run using either Docker/Podman, Apptainer/Singularity²⁰ or our own Tiny Range virtualisation implementation.²¹

Packaging software into Neurocontainers enables a modular deployment architecture, allowing for multiple software versions to co-exist on a single system. This design resolves issues related to dependency conflicts and version drift, allowing researchers to precisely match the software configuration used in original analyses. Neurocontainers are lightweight, portable, and can be version-controlled, enabling researchers to document, archive and recreate the exact computational software environment associated with a given analysis code for a given study. This approach also facilitates integration with reproducible workflow systems such as Jupyter Notebooks, Bash and Python scripts orchestrated through Neurocommand, and can be embedded into teaching materials, preprints, or publications to support re-execution by third parties.

IMPROVING ATTRIBUTION FOR CODE AND COMPUTATIONAL ENVIRONMENTS

To promote proper attribution and credit for software contributions and workflow development, Neurodesk implements a structured system for citation. All Neurocontainers are identified through a Digital Object Identifier (DOI), which captures the specific version of the software, down to the dependencies. These DOIs are registered and hosted through Zenodo and can be cited directly in scientific publications and workflows, allowing users to reference the specific Neurodesk environment set up for a given workflow, and reproduce the exact computational environment used for a specific tool in their analysis workflow.

The citation framework is aligned with the FORCE11 Software Citation Principles, which emphasise the importance of treating software as a legitimate and citable product of research.²² Neurodesk encourages users to cite both the overarching platform, the individual containers and the original developers of the tool, thereby supporting more granular recognition of developer contributions and enhancing transparency in computational reporting. This

standardisation of citation practices promotes reproducibility by linking analyses to the exact software environments used, while also strengthening traceability, incentivising proper credit, and fostering sustainable software development within the neuroscience community.

In a similar manner, Neurodesk offers users the ability to save and publish complete analysis workflows—including code, tool references, and metadata—and assign a DOI through Zenodo. This allows individual tools and full processing pipelines to be versioned, cited, and reused, supporting more granular attribution and improving the traceability of research outputs ([Figure 1](#)). By enabling citation of both containerised tools and structured workflows, Neurodesk reinforces software and methods as first-class research outputs.

POSITIONING NEURODESK WITHIN THE NEUROIMAGING ECOSYSTEM

Neurodesk complements, extends and interoperates with existing container solutions such as Neurodocker,¹⁸ and BIDS Apps,¹² or fMRIPrep,²³ by offering a general-purpose, cross-operating-system platform for managing a broad range of neuroimaging tools packaged in software containers. BIDS Apps provide containerised workflows for specific tasks such as structural and functional preprocessing and they are typically oriented toward automated, non-interactive processing. Neurodesk includes various BIDS Apps and makes them accessible for users unfamiliar with software containers. Neurodesk also packages and distributes containerised applications, such as fMRIPrep.²³ In addition, Neurodesk also packages tools into containers that are not yet distributed as containers, such as ANTs,¹⁶ ITKSnap,²⁴ FSL,¹³ and many others through an extensible container registry. Unlike many existing solutions, Neurodesk functions seamlessly across Windows, macOS, and Linux systems, lowering infrastructure barriers to be executed across local machines, institutional HPCs and cloud platforms. This makes Neurodesk particularly valuable for collaborative, cross-platform research teams and for reproducibility in peer review settings. Neurodesk currently uses Neurodocker to build the containers and combines it with the pre-built environments with a dynamic, user-facing infrastructure that supports teaching, review, and exploratory workflows.

Neurodesk also emphasizes usability. Tools can be accessible through a single terminal command or a graphical interface such as JupyterLab or VSCode. This flexibility enables both scripted and interactive workflows, including exploratory analyses and teaching use cases.

Compared to platforms like NeuroLibre,²⁵ which focuses on publishing interactive Jupyter-based papers, and the ReScience C journal, which emphasises the replication of computational results in formal publications, Neurodesk is designed for broader day-to-day research use across the project lifecycle, from development to peer review. Unlike [BrainLife.io](#),¹⁰ which offers an integrated web-based access to curated applications for processing neuroimaging workflows online, Neurodesk provides greater flexibility by enabling researchers to run their workflows locally, on HPCs,

or in cloud environments, without being tied to a centralised platform and supports a broader toolchain while allowing complete control over custom environments and interactive use. To fulfil its general-purpose nature, combined with support for FAIR data platforms like OpenNeuro and tools like DataLad, Neurodesk allows researchers to link software environments with open datasets in a reproducible manner. Researchers can use Neurodesk to re-run BIDS App workflows, perform interactive code review, or construct new multi-tool pipelines with full version control and citation support.

Rather than competing with existing solutions, Neurodesk serves as a unifying infrastructure that brings together standardised tools, reusable environments, and interoperable interfaces to support transparent and reproducible neuroscience across the research lifecycle.

RESULTS

The following three examples illustrate how Neurodesk can support key moments in the research lifecycle: peer review, post-publication re-use, and publishing new tools. Each use case is paired with a corresponding figure that shows an applied instantiation of the scenario described. While we keep the use case descriptions in the main text general, the captions of [Figures 2–4](#) provide further technical and contextual detail to ground each example.

USE CASE 1: FACILITATING CODE REVIEW IN SCIENTIFIC PUBLISHING

Ensuring computational reproducibility during peer review is often limited by practical barriers: reviewers may be using a different OS, unable to install the necessary software due to a lack of administrative rights to install dependencies, may be unable to replicate the analytical environment, or resolve version conflicts ([Figure 2a](#)). This limits the feasibility of verifying results, even when analysis code and data are nominally shared. For example, a reviewer assessing an fMRI analysis pipeline might require exact versions of multiple tools and specific dependencies, such as FSL, ANTs or FreeSurfer, alongside specific Python libraries, which can be time-consuming and error-prone to configure. After the tedious process of recreating the environment, the reviewer may still encounter incompatibilities or installation failures when attempting to run the pipeline on a different OS.

Neurodesk addresses these challenges by providing containerised environments that encapsulate the full software stack and dependencies needed to run computational analyses ([Figure 2b](#)), independent of the hardware and software environments that the containers are run on. These containers ensure that the software environment can be reproduced across various operating systems (Windows, macOS, GNU/Linux) and computing infrastructures (local machines, cloud instances or HPC clusters). The containerised structure preserves the execution environment and can be cited using a DOI, while the workflow and data document the specific analysis code and inputs required for reviewing. Hence, to fully support reproducibility, authors can share

the workflow containing the analytical code and, where appropriate, example datasets through external repositories. Reviewers can launch the exact same software stack used by the author, run the shared workflow with example data, and inspect outputs and code behaviour interactively. This portable solution simplifies the review process, reduces technical barriers, and promotes broader adoption of standardised practices in scientific publishing.

USE CASE 2: ENABLING REPRODUCIBILITY AND METHODOLOGICAL EXTENSION

A major obstacle to scientific progress is the difficulty of reusing or extending computational analyses, whether re-running one's own code after time has passed or adapting published methods for a new study. Differences in OS, software versions, and hidden assumptions in analytical pipelines often prevent researchers from reproducing prior results or adapting published workflows for new studies ([Figure 3a](#)). For instance, the default Python version on Ubuntu 24.04 is Python 3.12, while macOS Sonoma ships with Python 3.9, and system-level packages or syntax dependencies may break across these versions. These technical mismatches not only hamper reproducibility but also slows cumulative knowledge building.

Neurodesk enables researchers to assign both the full computational environment and the analytical workflows needed for reproducibility ([Figure 3b](#)). Through containerisation, authors preserve the exact software tools, libraries, and versions used in their analyses, independent of local system configurations and insulated from system-level library path discrepancies. Workflows are created separately, ensuring transparency and flexibility for users. Subsequently, researchers can retrieve the Neurodesk containers to recreate the original environment, access the workflows and datasets, and easily rerun previously developed workflows. Because containers maintain internal file structures and environment variables, library paths remain consistent across systems. This separation of environment preservation and workflow sharing simplifies replication, facilitates methodological extension, and supports a more iterative, transparent, and efficient model of scientific progress.

USE CASE 3: SUPPORTING TOOL DEVELOPMENT AND ATTRIBUTION

Many research software tools, particularly in fields like neuroimaging, are developed by individual researchers or small teams without formal mechanisms for citation, version control, or long-term maintenance. As a result, valuable tools may remain inaccessible, poorly documented, or insufficiently credited, despite being essential to downstream research. Traditional publishing models have often struggled to accommodate the dynamic and evolving nature of research software.

Neurodesk provides a sustainable infrastructure for sharing and crediting research tools ([Figure 4](#)). Developers can containerise their software within a standardised environment, preserving the specific computational context required for correct execution and minimising the risk of

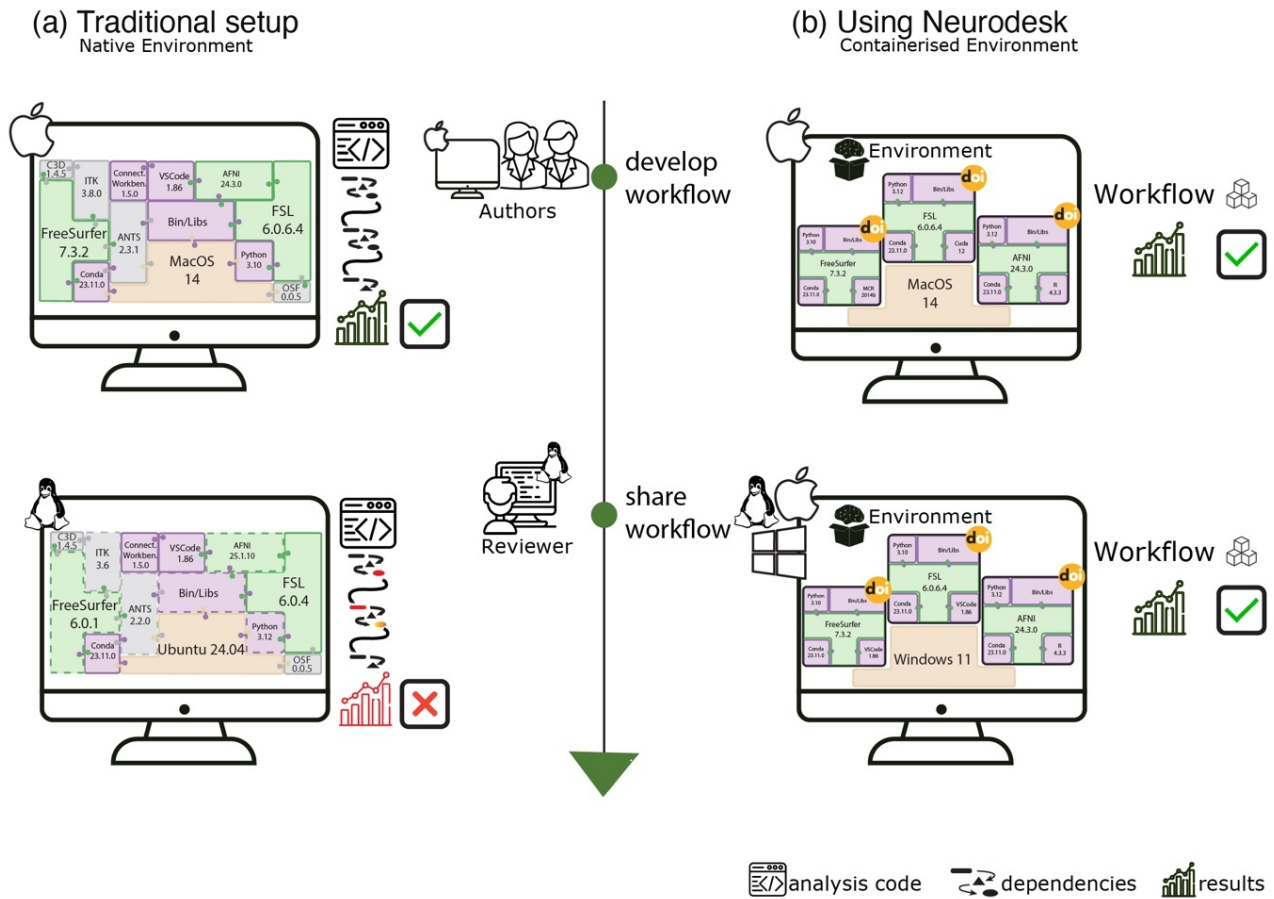


Figure 2. Use Case 1: This figure illustrates the challenges of verifying computational workflows across various operating systems during the peer-review process. In a traditional setup (a), the author develops an analysis script on macOS14 using FreeSurfer, FSL and AFNI. However, the reviewer using different operating systems (e.g., Ubuntu 24.04 Linux) may encounter incompatible dependencies or mismatched software versions, which can prevent them from running the analysis script and reproducing the results. This situation can lead to delays or discourage reviewers from evaluating the code altogether. In contrast, when using Neurodesk (b), the author shares the analysis code as a workflow along with citations for the containerised software environments (Neurocontainers) used. The reviewer can then reproduce the same analysis environment on any platform, Windows, macOS, or GNU/Linux, execute the workflow as intended, enabling code inspection and verification of results. By decoupling the execution environment from the host operating system, Neurodesk simplifies code review and enhances reproducibility in scientific publishing.

software breakage due to system incompatibilities. To lower the entry barrier for new contributors, Neurodesk offers both command-line templates and a user interface (UI) to streamline the creation of new containers (<https://neurodesk.org/neurocontainers-ui/>). Comprehensive developer documentation is available through the Neurodesk website (https://www.neurodesk.org/developers/new_tools/new_tool/), offering detailed guidance for updating existing containers or creating and submitting new tools.

Each container is assigned a persistent DOI through the platform’s integration with Zenodo, enabling formal citation in publications. This allows even small-scale contributions, such as a wrapper around a command-line tool or a bug fix in a preprocessing script, to be registered, reused, and cited. Developers can also share implemented workflows in their experiment, each assigned a DOI to enable proper citation and credit, thereby facilitating broader adoption by the community.

By decoupling the software tool from the host environment, providing version-controlled, executable containers, and embedding best practices for citation and metadata, Neurodesk advances efforts to treat research software as a first-class, citable scientific output, fully aligned with FAIR4RS and the FORCE11 Software Citation Principles.

DISCUSSION

The conventional frameworks of scientific publishing have not effectively addressed the reproducibility of computational research and the fair recognition of software development efforts. Unless these issues are resolved, the current “reproducibility crisis” is likely to continue, with important contributions to scientific foundations potentially going unrecognised and underappreciated. By providing containerised, citable environments and clear paths for developers to share their tools and workflows, Neu-

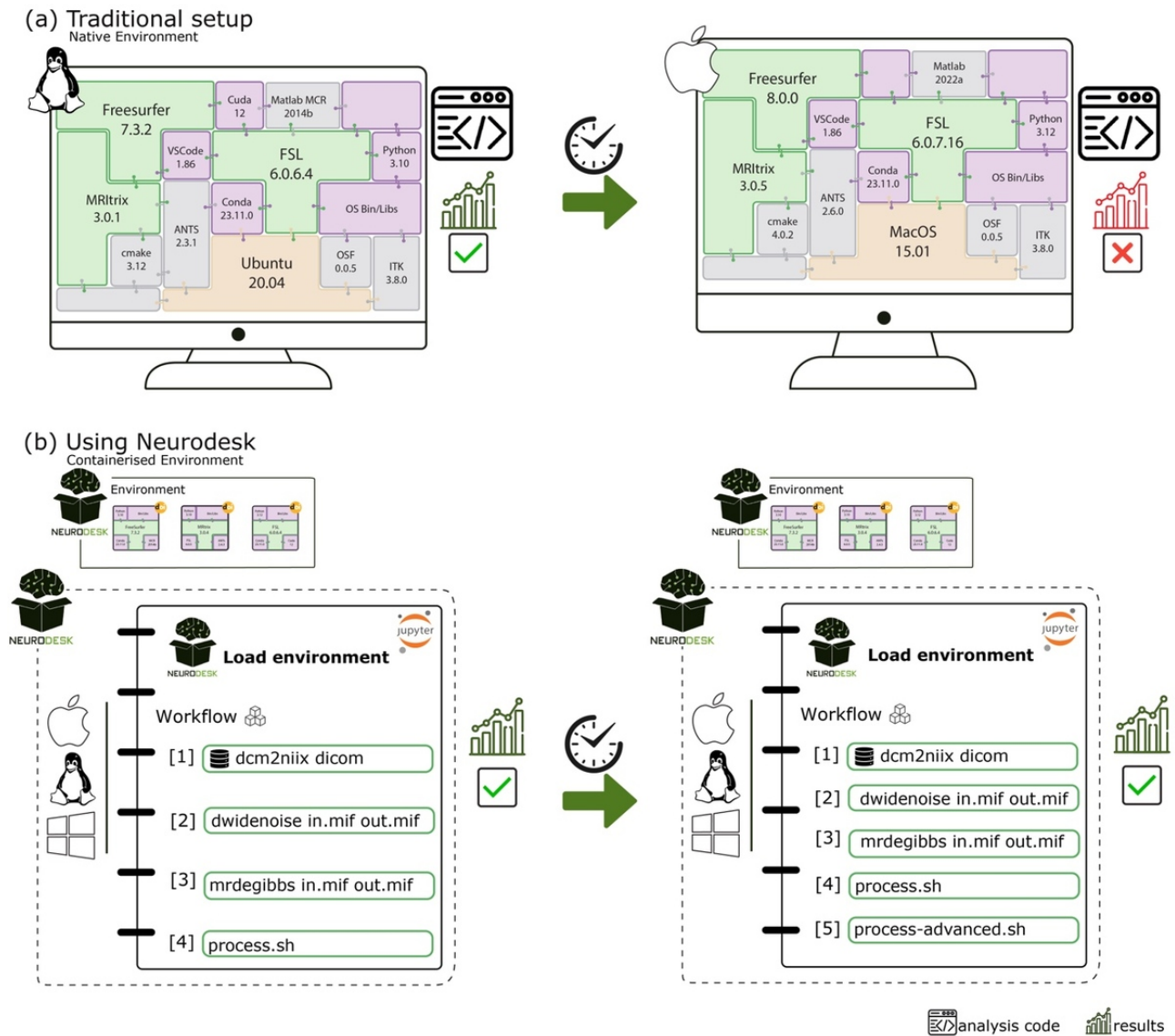


Figure 3. Use Case 2: A researcher plans to replicate and extend the diffusion MRI analyses developed and published during their PhD that used a complex pipeline involving MRtrix, FSL, and custom preprocessing scripts. The top row (a) illustrates a traditional scenario in which a researcher attempts to re-run the analysis script on a new computer in the native, uncontrolled environment. The original pipeline was run on Ubuntu 20.04 using specific software versions, but the researcher's new institutional machine runs macOS 15.01 with different versions of FreeSurfer, FSL and MRtrix, as well as different dependencies. Even though both systems include the required tools, differences in operating systems (Ubuntu vs macOS), system-level libraries, and subtle version mismatches lead to the script's failure. In contrast (b), using Neurodesk enables consistent re-execution of the original analysis across platforms. The researcher retrieves the exact Neurocontainers used during the original analysis, launches the JupyterLab interface, and loads the workflow with pre-mounted BIDS data. This environment includes the correct versions of MRtrix, FSL, and FreeSurfer, enabling the script to execute without errors. The researcher successfully replicates the original output and modifies the workflow to incorporate an updated processing. Neurodesk thus enables reproducibility across platforms and supports method reuse and extension, even years after the original project.

rodesk addresses key technical and cultural barriers to reproducibility and credit. It demonstrates how initiatives rooted in open science principles can promote greater transparency, traceability, sustainability, and recognition within the research ecosystem. To create a more equitable and replicable future for scientific publishing, it is essential to fully incorporate research software as a primary scholarly output.

To complement more extensive structural modifications, it is essential to develop concrete resources that facilitate the integration of open science concepts into everyday research routines. Through the examples presented, we have demonstrated how Neurodesk can: 1) facilitate objective peer review through executable code review; 2) enable the reproducibility and extension of published analyses through preserved computational environments; and 3)

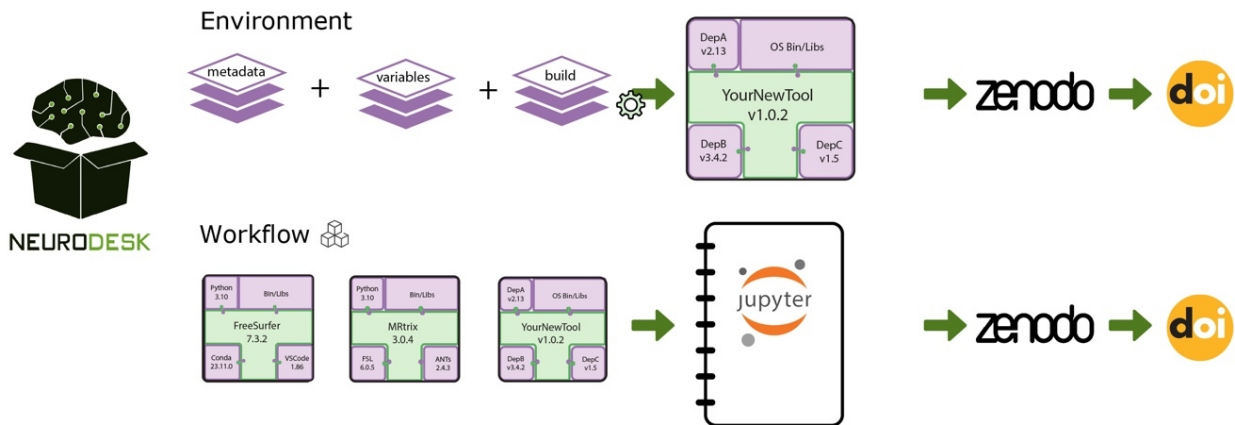


Figure 4. Use case 3: A research group develops a new automated hippocampal segmentation tool for high-resolution MRI scans. To ensure the tool is accessible and properly credited, they follow the Neurodesk developer guidelines available on the platform’s website to containerise their software. After successful integration into the Neurodesk ecosystem, the container is published to Zenodo to obtain a DOI. This container includes all software dependencies, ensuring consistent performance across systems and simplifying installation for end users and can now be loaded to any Neurodesk instance. They then create an interactive example Notebook demonstrating the intended workflow of the new tool and its interaction with other established tools like FreeSurfer and MRtrix. The workflow is preserved as a Jupyter notebook and published with a separate DOI, allowing other researchers to reproduce, test, and adapt the pipeline. In the manuscript describing the tool, the researchers reference the container DOI to credit the tool developers and the notebook DOI to credit workflow contributors. In future work, other researchers can easily download the container, access the documentation and example data, and run the tool reproducibly across platforms. The separation of environment (software container) and workflow (executable notebook) ensures long-term usability and transparency. This streamlined process enhances the tool’s visibility, ensures sustainable access, and provides formal academic recognition for the developers’ contribution.

support the sustainable publication and citation of new research tools and workflows. The platform operationalises these goals through a modular architecture, seamless integration with data- and workflow-sharing platforms, and accessibility features that lower the barrier to adoption. Rather than functioning as a specialised pipeline, Neurodesk offers a general-purpose, scalable platform that unifies containerised environments, data integration, and citation infrastructure.

CULTURAL ADOPTION AND USABILITY

While technical infrastructure is essential, the challenges of reproducibility are equally cultural. Adoption of new tools depends not only on their availability but also on their ease of use and demonstrated utility in real-world research workflows. Neurodesk is designed to reduce the friction typically associated with container-based systems by providing both graphical and command-line interfaces that work across Windows, macOS, Linux, and HPC systems. Moreover, Neurodesk can be installed on computers where users do not have administrative permissions. This cross-platform accessibility enables researchers, regardless of their prior experience with software containers, to easily launch tools, rerun workflows, and engage in code review without complex setup procedures.

To support uptake by users with varying technical backgrounds, Neurodesk is routinely integrated into teaching materials, online workshops, and university-level neu-

roimaging courses. For example, an interactive version of *Andy’s Brain Book*, one of the most widely used educational resources in neuroimaging, is currently being developed within the Neurodesk environment, allowing learners to interactively run code examples using pre-configured software stacks. Informal feedback suggests that students and early-career researchers are able to incorporate Neurodesk into their analysis workflows with minimal onboarding, supported by open-access documentation and tutorial notebooks: <https://neurodesk.org/example-notebooks/intro.html>.

As of mid-2025, Neurodesk has more than 1,500 monthly active users and is used by labs across Europe, North America, Asia, and Australia. The growing community participation and increasing number of shared workflows and citations suggest a promising trajectory toward broad adoption. Future work will include collecting structured usability feedback to inform further development and support sustained, community-driven growth.

THE IMPORTANCE OF ATTRIBUTION

Despite the central role of computational methods in modern science, contributions such as software development, infrastructure building, and documentation are often overlooked within traditional academic publishing models. Citations remain the primary metric of academic productivity and career advancement; yet, research software does not yet receive the formal recognition afforded to articles and

datasets. Developing and maintaining research software demands significant intellectual, technical, and collaborative effort, but these contributions typically fall outside the boundaries of conventional publishing formats. Without appropriate attribution mechanisms, the sustainability and innovation potential of research software ecosystems are undermined.

MEASURING THE IMPACT OF RESEARCH SOFTWARE

As scientific publishing continues to evolve, alternative metrics are needed to capture the true impact of software and infrastructure contributions. Traditional citation counts provide limited visibility into how widely software tools are used, adapted, or integrated into subsequent research. Complementary metrics such as the number of container downloads, GitHub repository forks and stars, DOI-based software citations, and documented use in independent research projects or clinical applications provide a broader view of scholarly influence. Tracking the incorporation of software into educational resources and training materials further highlights its dissemination and impact. Encouraging journals, funding agencies, and institutions to formally recognise these alternative impact measures is essential for building a more inclusive and representative system of academic evaluation.

RETHINKING ACADEMIC REWARD SYSTEMS

Ensuring equitable recognition of software contributions requires moving beyond traditional citation-based models through coordinated efforts across institutions, funding agencies, and journals. Policies must be updated to explicitly value research software, infrastructure contributions, and documentation efforts as legitimate scholarly outputs. Expecting researchers to ensure proper traceability and compliance with open science principles requires both institutional support and enforcement mechanisms. Evaluation criteria should integrate complementary metrics, such as software citations, repository activity, and usage in research and education, alongside traditional publications. However, a persistent limitation in current citation practices is the non-transitivity of credit: for example, when researchers build on top of existing pipelines or environments, citation may only be given to the top layer, with underlying infrastructure going uncredited. While Neurodesk provides citation metadata and encourages formal attribution at the container level, it cannot enforce downstream citations, and the responsibility remains with the user. This underscores the need to promote a broader ecosystem of credit that includes telemetry, usage statistics, and inclusion in peer-reviewed workflows. By establishing clear standards and incentives for recognising research software, the academic community can better support sustainable open science practices and ensure that critical contributions to the research ecosystem receive the credit they deserve.

SUSTAINABILITY AND LONG-TERM MAINTENANCE

Ensuring the long-term sustainability of software platforms requires more than initial funding, it also demands distributed maintenance, governance, and institutional partnerships. Neurodesk is supported by a sustainability model that includes multi-year infrastructure grants (e.g., Australian Research Data Commons (ARDC), the National Imaging Facility (NIF)), cross-institutional cloud hosting (e.g., Nectar Cloud, EGI, AWS, JetStream2), and delivery through a managed software-as-a-service model via the Queensland Cyberinfrastructure Foundation (QCIF). The Neurodesk platform is actively maintained by an open contributor community and hosted across federated research infrastructure providers, including the ARDC Nectar Cloud, Open Science Grid, EGI and JetStream2. These combined efforts aim to ensure the long-term accessibility, scalability, and impact of Neurodesk beyond the scope of individual grants.

NEURODESK'S ROLE IN SETTING A PRECEDENT

Initiatives such as Neurodesk can help set important precedents for the future of research attribution in neuroscience and beyond. By assigning persistent DOIs to software containers and workflows, supporting clear contribution pathways for developers, and promoting best practices for software citation and documentation, Neurodesk exemplifies how infrastructure projects can actively foster a culture of recognition.

As an open-source, cross-platform environment, Neurodesk complements task-specific tools and pipelines, acting as a backbone for reproducibility in neuroimaging across analysis types and a catalyst for attribution. By aligning with initiatives such as FAIR4RS and the FORCE11 Software Citation Principles, and by encouraging transparent practices across its ecosystem, Neurodesk offers a model for how future scientific publishing frameworks can more inclusively and sustainably value research software contributions.

To further support adoption, future efforts will focus on building partnerships with journals and editorial boards to facilitate migration of the containerised workflows in publications and facilitate reproducibility checks, as well as expanding onboarding materials and improving community documentation based on structured user feedback. These steps aim to ensure Neurodesk serves as infrastructure and actively contributes to shaping more reproducible publication practices in neuroimaging.

CODE AVAILABILITY STATEMENT

Open-source code for the development of Neurodesk infrastructure is available on GitHub (<https://github.com/neurodesk>).

ACKNOWLEDGMENTS

This work is supported by the Wellcome Trust with a Discretionary Award as part of the Chan Zuckerberg Initiative (CZI), The Kavli Foundation, and Wellcome's Essential Open Source Software for Science (Cycle 6) Program (Grant Ref: [313306/Z/24/Z]). This research was supported by the use of the ARDC Nectar Research Cloud, a collaborative Australian research platform supported by the Australian Research Data Commons (ARDC), a capability funded through the National Collaborative Research Infrastructure Strategy (NCRIS). This work benefited from services and resources provided by the EGI Federation with the dedicated support of CESNET-MCC. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech

Republic. This research was supported by Jetstream2 (NSF award #2005506), which is supported by the National Science Foundation. Jetstream2 is a cloud computing resource managed by the Indiana University Pervasive Technology Institute and part of the ACCESS project. The authors acknowledge the facilities and scientific and technical assistance of the National Imaging Facility, a NCRIS capability at the University of Queensland.

CONFLICTS OF INTEREST

The authors declare no competing interests.

Submitted: May 29, 2025 CDT. Revised: August 11, 2025 CDT.

Accepted: August 18, 2025 CDT. Published: September 19, 2025 CDT.



This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CCBY-4.0). View this license's legal deed at <http://creativecommons.org/licenses/by/4.0> and legal code at <http://creativecommons.org/licenses/by/4.0/legalcode> for more information.

REFERENCES

1. Wilkinson MD, Dumontier M, Aalbersberg IJ, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*. 2016;3:160018. doi:[10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18)
2. Chue Hong NP, Katz DS, Barker M, et al. *FAIR Principles for Research Software (FAIR4RS Principle)*. Zenodo; 2022. doi:[10.15497/RDA00068](https://doi.org/10.15497/RDA00068)
3. Sharma NK, Ayyala R, Deshpande D, et al. Analytical code sharing practices in biomedical research. *PeerJ Comput Sci*. 2024;10:e2066. doi:[10.7717/peerj-cs.2066](https://doi.org/10.7717/peerj-cs.2066)
4. Samuel S, Mietchen D. Computational reproducibility of Jupyter notebooks from biomedical publications. *Gigascience*. Published online January 2, 2024;13. doi:[10.1093/gigascience/giad113](https://doi.org/10.1093/gigascience/giad113)
5. Russell PH, Johnson RL, Ananthan S, Harnke B, Carlson NE. A large-scale analysis of bioinformatics code on GitHub. *PLoS One*. 2018;13(10):e0205898. doi:[10.1371/journal.pone.0205898](https://doi.org/10.1371/journal.pone.0205898)
6. Gomes DGE, Pottier P, Crystal-Ornelas R, et al. Why don't we share data and code? Perceived barriers and benefits to public archiving practices. *Proc Biol Sci*. 2022;289(1987):20221113. doi:[10.1098/rspb.2022.1113](https://doi.org/10.1098/rspb.2022.1113)
7. Barnes N. Publish your computer code: it is good enough. *Nature*. 2010;467(7317):753. doi:[10.1038/467753a](https://doi.org/10.1038/467753a)
8. Karakuzu A, DuPre E, Tetrel L, et al. NeuroLibre: A preprint server for full-fledged reproducible neuroscience. *OSFPreprint*. Published online 2022. doi:[10.31219/osf.io/h89js](https://doi.org/10.31219/osf.io/h89js)
9. Rougier NP, Hinsén K. ReScience C: A Journal for Reproducible Replications in Computational Science. Presented at: Reproducible Research in Pattern Recognition: Second International Workshop, RRPR 2018, Beijing, China, August 20, 2018, Revised Selected Papers; 2018; Beijing, China. doi:[10.1007/978-3-030-23987-9_14](https://doi.org/10.1007/978-3-030-23987-9_14)
10. Hayashi S, Caron BA, Heinsfeld AS, et al. brainlife.io: a decentralized and open-source cloud platform to support neuroscience research. *Nat Methods*. 2024;21(5):809-813. doi:[10.1038/s41592-024-02237-2](https://doi.org/10.1038/s41592-024-02237-2)
11. Renton AI, Dao TT, Johnstone T, et al. Neurodesk: an accessible, flexible and portable data analysis environment for reproducible neuroimaging. *Nat Methods*. 2024;21(5):804-808. doi:[10.1038/s41592-023-02145-x](https://doi.org/10.1038/s41592-023-02145-x)
12. Gorgolewski KJ, Alfaro-Almagro F, Auer T, et al. BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. *PLoS Comput Biol*. 2017;13(3):e1005209. doi:[10.1371/journal.pcbi.1005209](https://doi.org/10.1371/journal.pcbi.1005209)
13. Jenkinson M, Beckmann CF, Behrens TE, Woolrich MW, Smith SM. FSL. *Neuroimage*. 2012;62(2):782-790. doi:[10.1016/j.neuroimage.2011.09.015](https://doi.org/10.1016/j.neuroimage.2011.09.015)
14. Smith SM, Jenkinson M, Woolrich MW, et al. Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage*. 2004;23 Suppl 1:S208-19. doi:[10.1016/j.neuroimage.2004.07.051](https://doi.org/10.1016/j.neuroimage.2004.07.051)
15. Woolrich MW, Jbabdi S, Patenaude B, et al. Bayesian analysis of neuroimaging data in FSL. *Neuroimage*. 2009;45(1 Suppl):S173-86. doi:[10.1016/j.neuroimage.2008.10.055](https://doi.org/10.1016/j.neuroimage.2008.10.055)
16. Tustison NJ, Cook PA, Holbrook AJ, et al. The ANTsX ecosystem for quantitative biological and medical imaging. *Sci Rep*. 2021;11(1):9068. doi:[10.1038/s41598-021-87564-6](https://doi.org/10.1038/s41598-021-87564-6)
17. Tournier JD, Smith R, Raffelt D, et al. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *Neuroimage*. 2019;202:116137. doi:[10.1016/j.neuroimage.2019.116137](https://doi.org/10.1016/j.neuroimage.2019.116137)
18. ReproNim/neurodocker: 2.0.0. Published online 2025. doi:[10.5281/zenodo.14957539](https://doi.org/10.5281/zenodo.14957539)
19. Kennedy DN, Abraham SA, Bates JF, et al. Everything Matters: The ReproNim Perspective on Reproducible Neuroimaging. *Front Neuroinform*. 2019;13:1. doi:[10.3389/fninf.2019.00001](https://doi.org/10.3389/fninf.2019.00001)
20. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. *PLoS One*. 2017;12(5):e0177459. doi:[10.1371/journal.pone.0177459](https://doi.org/10.1371/journal.pone.0177459)
21. Scarsbrook J, Goddard E, Narayanan A, Dao T, Bollmann S. Neurodesk Portable: Containerized Neuroimaging across platforms with no installation. Published online 2025.

22. Smith AM, Katz DS, Niemeyer KE. Software citation principles. *PeerJ Computer Science*. 2016;2. doi:[10.7717/peerj-cs.86](https://doi.org/10.7717/peerj-cs.86)

23. Esteban O, Markiewicz CJ, Blair RW, et al. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat Methods*. 2019;16(1):111-116. doi:[10.1038/s41592-018-0235-4](https://doi.org/10.1038/s41592-018-0235-4)

24. Yushkevich PA, Piven J, Hazlett HC, et al. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*. 2006;31(3):1116-1128. doi:[10.1016/j.neuroimage.2006.01.015](https://doi.org/10.1016/j.neuroimage.2006.01.015)

25. Harding RJ, Bermudez P, Bernier A, et al. The Canadian Open Neuroscience Platform-An open science framework for the neuroscience community. *PLoS Comput Biol*. 2023;19(7):e1011230. doi:[10.1371/journal.pcbi.1011230](https://doi.org/10.1371/journal.pcbi.1011230)