Neurodesk: An accessible, flexible, and portable data analysis environment for reproducible neuroimaging

Angela I. Renton[1,2,#], Thanh Thuy Dao[2,#], David F. Abbott[15], Saskia Bollmann[2], Megan E. J. Campbell[18,19], Jeryn Chang[25], Thomas G. Close[16], Korbinian Eckstein[2], Gary F. Egan[13,17], Stefanie Evas[8,9], Kelly G. Garner[1,6], Marta I. Garrido[22,23], Anthony J. Hannan[15], Renzo Huber[20], Tom Johnstone[7], Jakub R. Kaczmarzyk[3,14], Lars Kasper[21], Levin Kuhlmann[27], Kexin Lou[2], Paris Lyons[7], Jason B. Mattingley[1,6], Akshaiy Narayanan, Franco Pestilli[26], Aina Puce[28], Fernanda L. Ribeiro[2], Nigel C. Rogasch[4,12,13], Thomas B Shaw[2,5,11], Paul F. Sowman[10], Gershon Spitz[24,29], Ashley Stewart[2], Ryan P. Sullivan[16], David J. White[7], Xincheng Ye[2], Judy D. Zhu[10], Aswin Narayanan[5,&], Steffen Bollmann[2,&]

- shared first-authorship

& - shared last-authorship

1. The University of Queensland, Queensland Brain Institute, St Lucia 4072, Australia.
2. The University of Queensland, School of Information Technology and Electrical Engineering, St Lucia 4072, Australia.
3. Medical Scientist Training Program, Stony Brook University, Stony Brook, NY, United States of America.
4. Discipline of Psychiatry, Adelaide Medical School, University of Adelaide, Australia
5. The University of Queensland, Centre for Advanced Imaging, St Lucia 4072, Australia.
6. The University of Queensland, School of Psychology, St Lucia 4072, Australia

7. Centre for Mental Health & Brain Sciences, Swinburne University of Technology, Hawthorn 3122, Australia
8. School of Psychology, University of Adelaide, Adelaide, 5000, Australia
9. Human Health, Health & Biosecurity, CSIRO, Adelaide, 5000, Australia
10. Macquarie University, School of Psychological Sciences, North Ryde 2112, Australia.
11. Department of Neurology, Royal Brisbane and Women's Hospital, Brisbane Australia
12. Hopwood Centre for Neurobiology, Lifelong Health Theme, South Australian Health and Medical Research Institute (SAHMRI), Adelaide, SA, Australia
13. The Turner Institute for Brain and Mental Health, School of Psychological Sciences, Monash University, Victoria, Australia
14. Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, United States of America
15. The Florey Institute of Neuroscience and Mental Health, The University of Melbourne, Victoria, Australia
16. The University of Sydney, School of Biomedical Engineering, Sydney, Australia
17. Monash Biomedical Imaging, Monash University, Victoria, Australia
18. School of Psychological Sciences, University of Newcastle, Australia
19. Hunter Medical Research Institute Imaging Centre, Newcastle, Australia
20. Functional Magnetic Resonance Imaging Core Facility (FMRIF), National Institute of Mental Health (NIMH), USA
21. Techna Institute, University Health Network, Toronto, Canada
22. Melbourne School of Psychological Sciences, The University of Melbourne
23. Graeme Clark Institute for Biomedical Engineering, The University of Melbourne
24. Department of Neuroscience, Central Clinical School, Faculty of Medicine, Nursing and Health Sciences, Monash University, Australia.
25. The University of Queensland, School of Biomedical Sciences, St Lucia 4072, Australia
26. Department of Psychology, Center for Perceptual Systems, Center for Theoretical and Computational Neuroscience, Center on Aging and Population Sciences, Center for Learning and Memory, The University of Texas at Austin, 108 E Dean Keeton St, Austin, TX 78712, USA
27. Department of Data Science and AI, Faculty of Information Technology, Monash University, Clayton VIC 3800, Australia.
28. Department of Psychological and Brain Sciences, Indiana University, Bloomington, IN 47405, USA.
29. Monash-Epworth Rehabilitation Research Centre, Turner Institute for Brain and Mental Health, School of Psychological Sciences, Monash University, Clayton, 3168, Australia.

Corresponding Author

Steffen Bollmann (s.bollmann@uq.edu.au)

Acknowledgements

This project's development was driven by investment from the Australian Research Data Commons (ARDC) through the Australian Electrophysiology Data Analytics PlaTform (AEDAPT). Thank you to Oracle for Research for providing Oracle Cloud credits and related cloud resources to support this project. The University of Queensland funded the project via the Knowledge Exchange & Translation (Kx&T) Fund and via the UQ AI Collaboratory. The authors acknowledge the facilities and scientific and technical assistance of the National Imaging Facility, a National Collaborative Research Infrastructure Strategy (NCRIS) capability.

Author contributions (CRediT statement[1])

***Conceptualization*:** Steffen Bollmann, Aswin Narayanan, Tom Johnstone, David White, Angela Renton, Thomas Shaw, Ryan Sullivan, Thomas Close, Anthony Hannan, Gary Egan, Marta Garrido, Aina Puce, Franco Pestilli, Levin Kuhlmann, Gershon Spitz, David Abbott, Megan Campbell, Nigel Rogasch, Jakub Kaczmarzyk. ***Software*:** Steffen Bollmann, Aswin Narayanan, Thomas Shaw, Tom Johnstone, David White, Akshaiy Narayanan, Thuy Dao, Ashley Stewart, Lars Kasper, Judy D Zhu, Korbinian Eckstein, Stefanie Evas, Xincheng Ye, Fernanda Ribeiro, Jeryn Chang, Kexin Lou, Renzo Huber. ***Validation*:** Steffen Bollmann, Aswin Narayanan, Angela Renton, Thomas Shaw, David White, Kelly Garner, Thuy Dao, Ashley Stewart, Lars Kasper, Judy D Zhu, Korbinian Eckstein, Stefanie Evas, Xincheng Ye, Fernanda Ribeiro, Jeryn Chang, Jakub Kaczmarzyk, Kexin Lou, Renzo Huber, Ryan Sullivan, Thomas Close, Levin Kuhlmann, Gershon Spitz, David Abbott, Megan Campbell, Nigel Rogasch, Marta Garrido, Aina Puce. ***Formal analysis*:** Thuy Dao. ***Conceptualisation of Formal analysis*:** Steffen Bollmann, Thuy Dao, Angela Renton, Fernanda Ribeiro, Thomas Shaw. ***Writing - Original Draft*:** Angela Renton. ***Writing - Review & Editing*:** All Authors. ***Visualisation*:** Angela Renton. ***Supervision*:** Steffen Bollmann, Tom Johnstone, Angela Renton. ***Project administration*:** Steffen Bollmann, Aswin Narayanan, Paris Lyons, Tom Johnstone. ***Funding acquisition*:** Steffen Bollmann, Aswin Narayanan, Tom Johnstone, David White.

Competing interests

The authors declare no financial interests

Abstract

Neuroimaging data analysis often requires purpose-built software, which can be difficult to install and may produce different results across computing environments. Beyond being a roadblock to neuroscientists, these issues of accessibility and portability can hamper the reproducibility of neuroimaging data analysis pipelines. Here, we introduce the Neurodesk platform, which offers a sustainable, flexible solution; harnessing software containers to support a comprehensive and growing suite of neuroimaging software (https://www.neurodesk.org/). Neurodesk includes both a browser-accessible virtual desktop environment and a command line interface, mediating access to containerised neuroimaging software libraries from multiple systems; including personal computers, cloud computing, high-performance computers, and Jupyter notebooks. This community-driven, open-source platform represents a paradigm shift for neuroimaging data analysis, allowing for accessible, fully reproducible and portable data analysis pipelines, which can be redeployed in perpetuity, in any computing environment, with ease.

Introduction

Neuroimaging data analysis is a challenging enterprise. Aside from the neuroscientific principles motivating the choice of analysis, building an analysis pipeline requires advanced domain knowledge well beyond the researcher's own topic area; for example, signal processing, computer science, software engineering, statistics, and applied physics. Researchers faced with this daunting task typically rely on multiple, specialised software packages used in custom pipelines to suit a specific aim. These packages are often developed using a not-for-profit model by researchers with limited resources, and so have little dedicated technical support. As a result, packages are often difficult to install, and inconsistently supported across computing environments[2–4]. Consequently, researchers often limit themselves to fewer, often less advanced or out of date tools, and spend considerable time installing and compiling software, undermining both scientific productivity and reproducibility. To address these issues, we developedan open-source and community-oriented solution to enable neuroscientists to develop neuroimaging analysis workflows in line with four guiding principles: *Accessibility*, *Portability*, *Reproducibility*, and *Flexibility*.

Ideally, the software and code used in any scientific analysis workflow should be easily *accessible*, such that the workflow can be deployed without substantial investment of time or effort by users[5], and *portable*, such that analysis workflows can be tractably shifted between computing environments. Many researchers prototype analysis pipelines using their own local computers and later switch to workstations and high-performance computing clusters for processing their datasets at scale. Accessible and portable workflows therefore allow for optimised allocation of computing resources while supporting shared development workloads amongst collaborators[6]. Unfortunately, many neuroimaging data analysis workflows are neither readily accessible nor portable for scalable computing[7,8]. This is because many workflows rely on specialised tools purpose-built by a small number of skilled developers, often on short-term contracts, who are then burdened with the task of continuously adapting their tools to evolving computing environments[2].

Beyond the costs to productivity, the inaccessibility and instability of many neuroimaging tools poses a wider threat to *reproducibility*[9–16]. The transparency and openness promotion (TOP) guidelines, which to date have over 5,000 journals and

organisations as signatories, state that all reported results should be independently reproduced before publication[17]. In reality, this is impractical and too time consuming to implement at review[8]. Where analysis pipelines can be ported, subtle differences in the implementation of specific processing steps across computing environments can alter results[18–21]. Thus, it is often not possible to precisely reproduce the results of a prior study, even given the original data and analysis protocol[18].

Unfortunately, many existing solutions lack the required *flexibility* for research applications of neuroimaging data analysis[22]. For example, single-install pre-programmed analysis pipelines are a popular solution amongst clinicians, but researchers typically custom tailor analysis pipelines toward specific research questions[23–25]. The issues of inaccessibility in neuroimaging software have been recognised by the NeuroDebian[2] and NeuroFedora[26] projects, which provide a wide range of neuroimaging tools packaged for Linux operating systems. However, the majority of neuroscientists do not use Linux on their personal computers and thus still cannot easily access these packages[3]. To address this barrier, researchers often use dual boot computers or virtual machines. These solutions are resource intensive and force researchers to develop less flexible workflows due to the practical limitations inherent to installing new tools. While compiled packages make installations easier, applications still need to be installed on the host computer and suffer the usual problems of conflicts between different software packages, software versions, or the libraries they require to be installed (software "dependencies"). Many researchers are also limited in flexibility by institutional restrictions imposed on the installation of new software.

Applications with highly specific or conflicting dependencies are by no means unique to neuroscience. This universal issue has led to the development of software containers: lightweight, portable solutions for running and sharing individual applications. Software containers package specific applications along with their dependencies. Container engines such as Docker and Apptainer/Singularity allow containerised applications to run on various host operating systems and computing environments, while keeping separate containers isolated from each other and the host machine, eliminating concerns about conflicting or missing dependencies[27,28]. These benefits make software containers ideally suited to tackle the issues relating to the development of scientific analysis workflows described above[29]. However, despite the benefits of containerisation, only a small number of

integrated neuroscience-specific or adaptable workflow systems support containerised distributed computing[6,30–32]. While platforms such as OpenNeuro[33], Brainlife[34], Flywheel[35], XNAT[36] and Qmenta[37] have drastically improved the accessibility and reproducibility of Neuroimaging analyses, these platforms still lack portability. Indeed, no solution exists that universally addresses the issues raised above. Our objective is to change this with the development of Neurodesk: a community-driven open-source platform which harnesses software containers to create an accessible and portable data analysis environment that allows users to flexibly and reproducibly access a comprehensive range of neuroimaging tools through both a user-friendly graphical desktop and command line interface.

Results

Overview of the Neurodesk platform

Here we present Neurodesk; a platform facilitating *Accessibility, Portability, Flexibility,* and *Reproducibility* for Neuroimaging data analysis (**Figure 1**). In developing Neurodesk, we focussed strongly on the outcome of sustainability to ensure that workflows developed on the Neurodesk platform remained consistent with these four guiding principles across updates to users' local computing environments. In this section, we introduce the available tools in the Neurodesk platform, discuss how these address the issues raised above and report the results of an empirical evaluation of reproducibility in Neurodesk. For further details of the rationale behind the approaches adopted to achieve these results, please see the online methods.

At the core of Neurodesk are Neurocontainers; a collection of software containers that package a comprehensive and growing array of versioned neuroimaging tools (**Figure 1*b***). The build scripts for these software containers are stored in an open-access git repository. Using continuous integration tools, new container build scripts contributed by the community are automatically built as software containers and can be accessed throughout the Neurodesk platform or as standalone tools (**Figure 1*a***). Each individual 'Neurocontainer' includes the packaged tool as well as all of the dependencies required to execute that tool, allowing it to run on various computing architectures (**Figure 1*c***). As the containers isolate dependencies, different Neurocontainers can provide different versions of the same tool without conflicts. This allows researchers to seamlessly transition between different versions of software across projects, or even within a single analysis pipeline. To facilitate access to this software, we provide an accessibility layer, through which users can access software directly through the cloud or download containers for offline use, all without the need to install software or packages on the local system (**Figure 1*b***).

We provide two options for interfacing with Neurocontainers. The first is Neurodesktop, a remote-desktop and browser-accessible virtual desktop environment in which any of the containerised tools can be easily launched from the application menu (**Figure 1*d***). As such, analysing neuroimaging data through Neurodesktop has the look and feel of working on one's local computer. For more advanced users and for HPC environments, we developed Neurocommand, a tool for interfacing with Neurocontainers

through the command-line (**Figure 1*d***). Both of these interfaces can be deployed across almost[1] any computing hardware and modern operating systems, meaning that analysis pipelines developed using the Neurodesk platform are reproducible and can be scaled from local computers to cloud and HPC environments. Neurocontainers can even be used inside Jupyter notebooks, meaning that analysis pipelines developed using Neurodesk can be easily shared alongside published manuscripts (**Figure 1*d***).

[1] N.B. At the time of publication, Neurodesk is not supported for the ARM processors equipped in M1 Mac computers. However, this is an area of active development for the Neurodesk team that will be addressed in a future release.

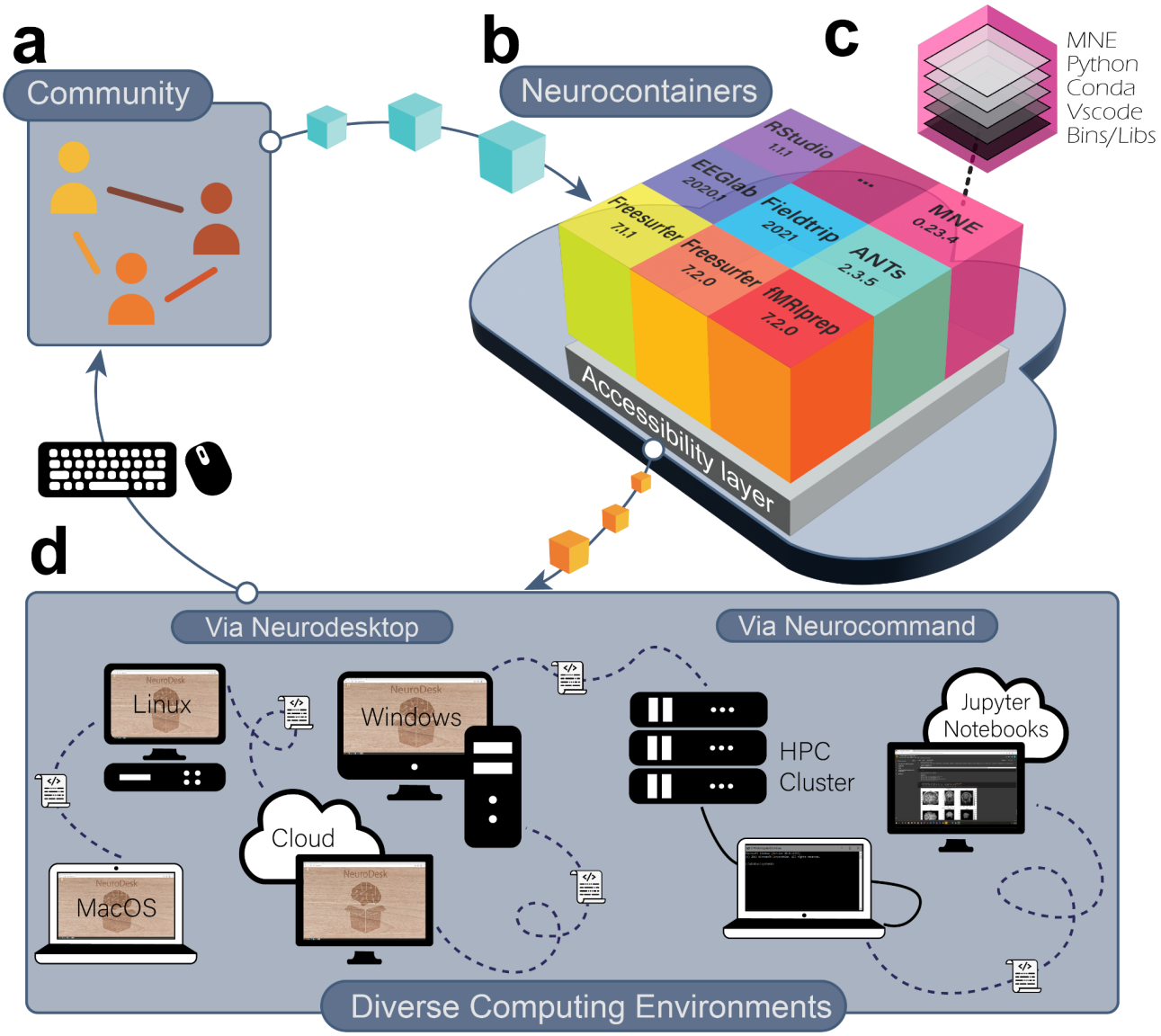


Figure 1. The Neurodesk platform. (**a**) The Neurodesk platform is built by, and for, the scientific community, such that anyone can contribute recipes for new software containers to the repository. (**b**) Recipes contributed by the community are automatically used to build software containers, which are stored in the Neurocontainers repository. (**c**) Each software container packages a tool together with all of the required runtime dependencies. This means that the packaged software is able to run identically in any supported computing environment. (**d**) Neurodesk provides two layers of accessibility through which the software containers in the Neurocontainers repository can be run: 1. Neurodesktop is a browser-accessible virtual desktop environment, allowing users to interact with the containerised software. 2. Neurocommand is a command-line interface, which allows users to run the same software containers in a programmatic way, suitable for HPC platforms and running software without a graphical user interface (GUI). Together, these interfaces allow users to reproduce the same analysis pipelines across a range of computing environments.

How to use Neurodesk: Accessibility, Flexibility & Portability

A core aim behind Neurodesk is to provide a platform that makes building and running reproducible analysis pipelines easily accessible to all researchers. The platform website (https://Neurodesk.org/) has been designed to be user-friendly and open to community contributions. This includes automatically updating information about the containerised software included with each new release directly from the Neurodesk repository. As such, users are continuously presented with up-to-date documentation, lists of currently available applications, and release history. The website hosts clear instructions and guidance for how to access and interact with Neurodesk from a variety of computing environments.

Besides ensuring that users always have access to thorough, clear, and up-to-date documentation, we have taken additional steps to ensure that Neurodesk makes reproducible neuroimaging data analysis *accessible*. Neurodesk can be accessed from almost any computing environment; because the tools have been containerised, they have access to exactly the same dependencies no matter where they are run. This mobility extends to the Neurodesktop graphical user interface (GUI), which provides the same desktop environment across all supported computing environments. This allows containerised analyses to look, feel, and run exactly the same way across all supported computing environments. Thus, researchers reading or reviewing manuscripts with open-source data and code can use Neurodesk to replicate the *exact* pipeline using the reported tool versions without having to install any additional software, thus avoiding the risk of interfering with existing versions of the tools that they use for their own data analysis.

For a data analysis environment to be *portable*, such that it can easily shift between computing environments, it also needs to be light-weight with a small storage footprint. To this end, our accessibility layer harnesses the CERN Virtual Machine File System (CVMFS)[38]. The CVMFS layer allows software to be accessed and run locally from a remote host without installation, such that only those parts of a container which are actively in use are sent over the network and cached on the user's local computer. Practically, this means that users can access terabytes of software without having to download or store it. The Neurodesk platform has a number of CVMFS nodes across the world providing low latency, direct access to Neurocontainers. Thus, to use Neurodesk, users need only to install the

required container engine and access the Neurocontainer of their choice. For Neurodesktop, which facilitates access to all tools in the Neurocontainers repository, the download is only ~1GB in total.

Anticipating that the installation of a third-party container engine software may be a barrier to entry for some researchers, we have developed an entirely cloud-based Neurodesktop service; 'Neurodesk Play' (http://play.neurodesk.org). Neurodesk Play is accessible globally, allowing anyone around the world to access a cloud-based graphical desktop environment for neuroimaging data analysis. While computing resources in Neurodesk Play are limited, Neurodesktop can also be hosted on institutional or cloud computing resources where more compute resources are available. For example, Neurodesktop is freely available to all publicly funded researchers in Australia and New Zealand on the Nectar Research Cloud Virtual Desktop Service provided by the Australian Research Data Commons (ARDC).

Long Term Sustainability of the Neurodesk Platform

Neurodesk has a wide selection of tools available spanning many domains of neuroimaging data analysis. **Table 1** shows the tools available at the time of publication, though this list is growing rapidly. A full up-to-date list can be found at https://Neurodesk.org/applications/. Neurodesk employs a two-pronged approach to staying up-to-date with new neuroimaging tools and new versions of already included software: a.) The Neurodesk maintainers add tools as they become aware of them, or from requests and contributions from the community. The Neurodesk GitHub repository (https://github.com/NeuroDesk) has an active discussion forum where developers respond to requests for new software containers. b.) In addition to this developer-centric route to new software containers, we actively encourage contributions from the research community. A core aim for the development of the Neurodesk platform was to develop a sustainable community-driven project that is not contingent on a specific team of developers. As such, we provide a template and detailed instructions for creating build scripts for new software containers. Using continuous integration and deployment, community contributed build scripts for new containers are automatically built, screened, and deployed with the daily release of Neurodesk to be accessed by the global community.

Table 1. Tools currently available in Neurodesk as of 21/12/22 (retrieved from https://Neurodesk.org/applications/). Note that each tool has been listed under only one category, though some may span multiple categories.

Category	Tool
Editors and Programming	VS Code, Gedit, Emacs, Vim, Python, Git, Julia, Matlab, ROOT, RStudio
System Management	Lmod, Singularity, Htop
Data Synchronisation Tools	Rsync, Rclone, Nextcloud client, Owncloud client, Globus personal connect
Browers and Networking	Firefox, OpenSSH client
Workflows	Nipype[39], ASLPrep[40], fMRIPrep[41], MRIQC[42], QSMxT[43]
Data Organisation	BIDScoin[44], BIDStools[45],Convert3D[46]
Diffusion MRI	Diffusion Toolkit[47], DSI Studio[48], MRtrix[49], MRtrix3Tissue[50], TrackVis[47]
Rodent Imaging	AIDAmri[51], RABIES[52]
Spectroscopy	LCModel[53], MRSIProc[54]
Structural and/or Functional Imaging	AFNI[55], ANTs[56], ASHS[57], BART[58], CAT12[59], CLEAR-SWI[60], CLEAR-SWI[60], Conn[61], Connectome Workbench[62], FatSegNet[63], FreeSurfer[64], FSL[65]FSL, HD-BET[66], LASHiS[67], LayNii[68], MINC[69], MRItools[70], NiftyReg[71], NiiStat[72], OSH-yX[73], Palm Alpha[74], PhysIO[75], ROMEO[76], Slicer[77], Spinal Cord Toolbox[78], SPM[79], TGVQSM[80]
Electroencephalography (EEG) and/or Magnetoencephalography (MEG)	Brainstorm[81], EEGLAB[82], FieldTrip[83], MNE[84], Sigviewer[85]
Machine Learning and Statistics	R[86], Deep Retinopy[87], Delphi[88]
Visualisation and Image Editing	ImageMagick[89], GIMP[90], itk-SNAP[46], MRIcron[91], MRIcroGL[92], SicerSALT[93], Surf Ice[94]

Reproducibility in Neurodesk

Scientific progress fundamentally depends on the peer review process, such that scientists must be able to critically assess reported findings and conclusions based on a clear and thorough methodological description[95]. Well-documented experimental code is the most thorough description of any analysis pipeline. However, differences in computing environments and dependencies mean that access to this source code does not guarantee the same result[96,97]. Reproducibility, defined as "running the same software on the same input data and obtaining the same result"[15,95,96], has therefore come to represent a minimum standard by which to judge scientific claims[15,95,96]. Unfortunately, scientific reproducibility is often not attainable due to differences in the outcomes of neuroimaging pipelines across different computing environments as previously documented[18,98,99]. Glatard et al. (2015) demonstrated this effect for several fMRI analysis pipelines, showing that differences in the implementation of floating-point arithmetic across operating systems accumulated over the course of long analysis pipelines, and led to meaningful differences in the results[18]. The Neurodesk platform solves this issue through its use of containerised software, which guarantees the same runtime dependencies across computing environments. To evaluate this claim, we replicated Glatard et al.'s analyses using Neurodesk vs locally installed software across different operating systems.

Methodological approach. The widely used FMRIB Software Library (FSL) 6.0.5.1[65] was installed both locally and within Neurodesk on two separate computers (System A, System B) running different Linux distributions, resulting in four unique computing environments (see **Table 2**). Glatard et. al's FSL-based analyses, namely the Brain Extraction Tool (FSL-BET; see online methods), tissue classification (FMRIB's Automated Segmentation Tool [FSL-FAST]), image registration (FMRIB's Linear Registration Tool [FSL-FLIRT]), and subcortical tissue segmentation (FMRIB's Integrated Registration and Segmentation Tool [FSL-FIRST]) were replicated in each of these environments using 157 T1-weighted magnetic resonance images (MRI) from the International Consortium for Brain Mapping (ICBM)[100]. Each analysis was run twice within each environment to verify that there was no intra-environment variability. To evaluate the reproducibility of the analysis environment using locally installed vs Neurodesk software, we compared the outputs for each installation type across computers (System A vs System B). For both intra- and inter-environment comparisons, we begin by comparing file "checksums"; alphanumeric

values which uniquely represent the contents of a file, such that two identical files will result in identical checksums. When two files produced different checksums, we quantified the pairwise differences across systems by computing dice similarity coefficients across images (**Figure 2a**). Note that there were never any intra-system differences in checksums (i.e., all analyses were determinative, resulting in identical outcomes when run twice in the same computing environment). The code used to implement these analyses is available and can be run through Neurodesk Play at: https://osf.io/e6pw3/.

Table 2. Computing environments used to run analyses.

	System A		System B	
	Local	Neurodesk	Local	Neurodesk
Applications	FSL 6.0.5.1	FSL 6.0.5.1	FSL 6.0.5.1	FSL 6.0.5.1
Glibc version	2.31	2.23	2.28	2.23
OS	Ubuntu 20.04	Ubuntu 16.04.7	AlmaLinux 8.5	Ubuntu 16.04.7
Hardware	12th Gen Intel(R) Core(TM) i7-12700		AMD EPYC 7542 32-Core Processor	

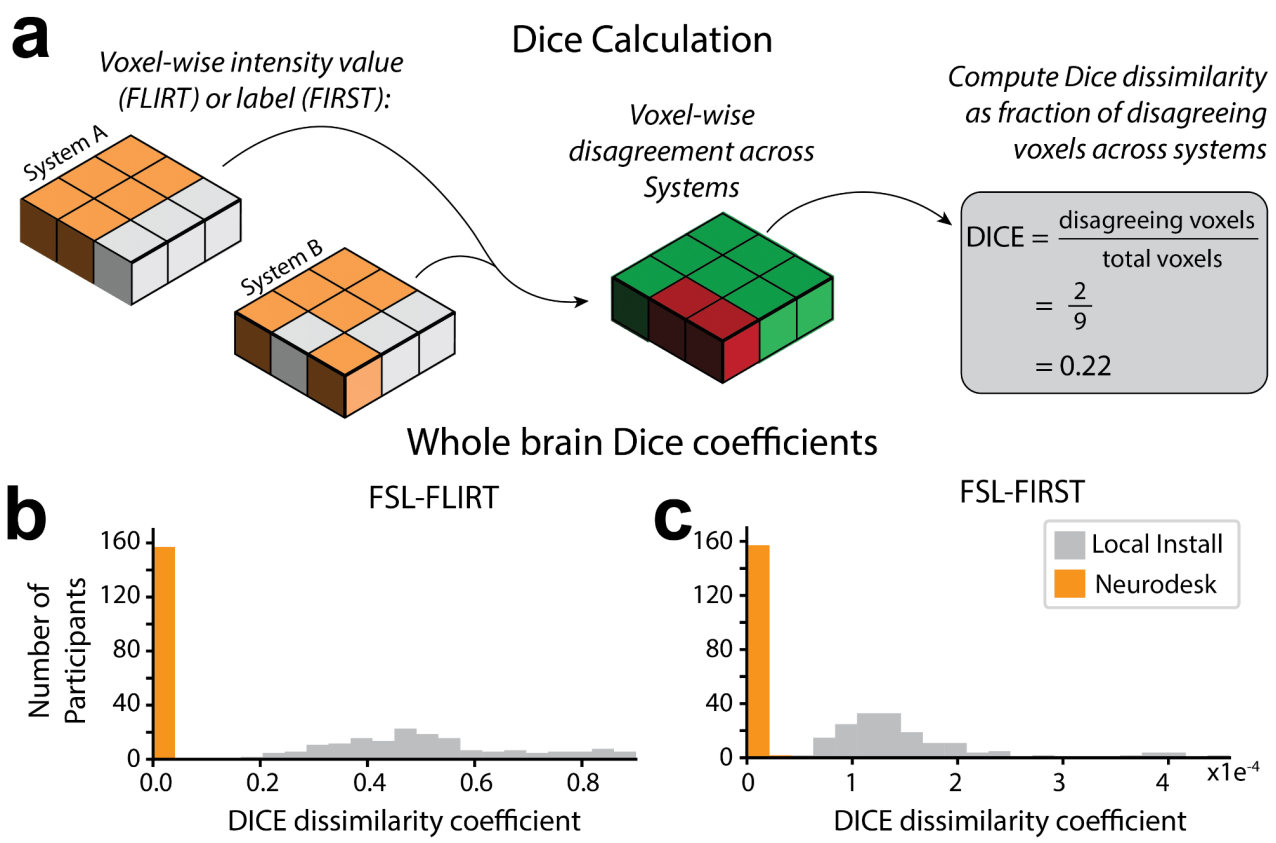


Figure 2. Discrepancies in image registration and tissue segmentation. (**a**) Calculation of the Dice dissimilarity coefficients; for each image, the voxel-wise disagreement in image intensity (FLIRT) or label (FIRST) calculated on System A vs System B was expressed as a proportion of the total number of voxels for each participant. (**b**) Histograms of Dice dissimilarity coefficients for image intensity calculated with FSL-FLIRT on Neurodesk vs. Local Install. To calculate these Dice coefficients, "disagreement" meant a voxel had a different intensity after image registration on System A vs. System B. Thus, the Dice coefficient of 0 for every single participant whose images were registered using Neurodesk means that the image intensity of each of these participants was perfectly matched across systems at every single voxel. (**c**) Histograms of Dice dissimilarity coefficients for subcortical structure labels calculated using FSL-First on Neurodesk vs. Local Install. To calculate these Dice coefficients, "disagreement" meant a voxel had different labels (e.g., amygdala, hippocampus, etc.) after image segmentation on System A vs. System B. Note that these Dice coefficients are, overall, much smaller than for image registration. This is to be expected as there are 238 times more "classes" for the image registration task than the classification task. Notably, however, while both Neurodesk and the local system show strong agreement across systems overall, these distributions are completely non-overlapping, with Neurodesk showing much greater reliability across systems.

Image registration. FSL FLIRT was applied to register the images to the standard MNI-152 T1 1 mm template using 12 degrees of freedom. When run through Neurodesk, the outputs of this processing step had identical file checksums across computing systems for all images. However, file checksums for local installations of FSL did not match across systems. Dice dissimilarity coefficients for each image were computed to quantify the pairwise differences in image intensity across systems (**Figure 2a**). Voxel-wise agreement in image registration for Neurodesk was perfect (Dice dissimilarity coefficient; Range: 0.00, M = 0.00, SD = 0.00). However, there were many voxels with differing intensity across local installations (Dice dissimilarity coefficient; Range: 0.19 – 0.90, M = 0.51, SD =0.17, **Figure 2b**). These high Dice dissimilarity coefficients for the local installation indicate differences across many voxels, however, the magnitude of these differences in image intensity was typically subtle (inter-system intensity difference; M = 1.88, SD = 1.97; where $intensity \in Z$: $intensity \in [0, 1903]$, **Figure 3a, b**).

Subcortical tissue segmentation. Differences in image intensity across local installations were widespread, but subtle. In line with Glatard et. al's approach, we next asked whether these differences impacted subcortical tissue segmentation (using FSL FIRST); the next step in the analysis pipeline. File checksums for the segmentation outputs matched for 0% of images when run using the local installation and 93% of images when run with Neurodesk. Computation of the Dice dissimilarity coefficients for each type of installation revealed that while differences were small, overall, they had non-overlapping ranges. Indeed, differences were much less prevalent for the Neurodesk installations (Dice dissimilarity coefficient; Range: 0.00 – 2.20×10^{-5}, M = 3.43×10^{-7}, SD < 0.01) compared with the local installations (Dice dissimilarity coefficient; Range: 5.80×10^{-5} – 4.59×10^{-4}, M = 1.46×10^{-4}, SD < 0.01, **Figure 2c**). Notably, this means that on average, there was 426x more voxel-wise disagreement across systems for the locally installed software than for neurodesk. This difference can be visualised by comparing the 3D projections of the mean inter-system differences in classification across participants (**Figure 3c, d**). These projections illustrate that differences for locally installed software were widespread and spanned across all subcortical structures (**Figure 3c**), while any subtle differences for Neurodesk were limited to a few voxels (**Figure 3d**).

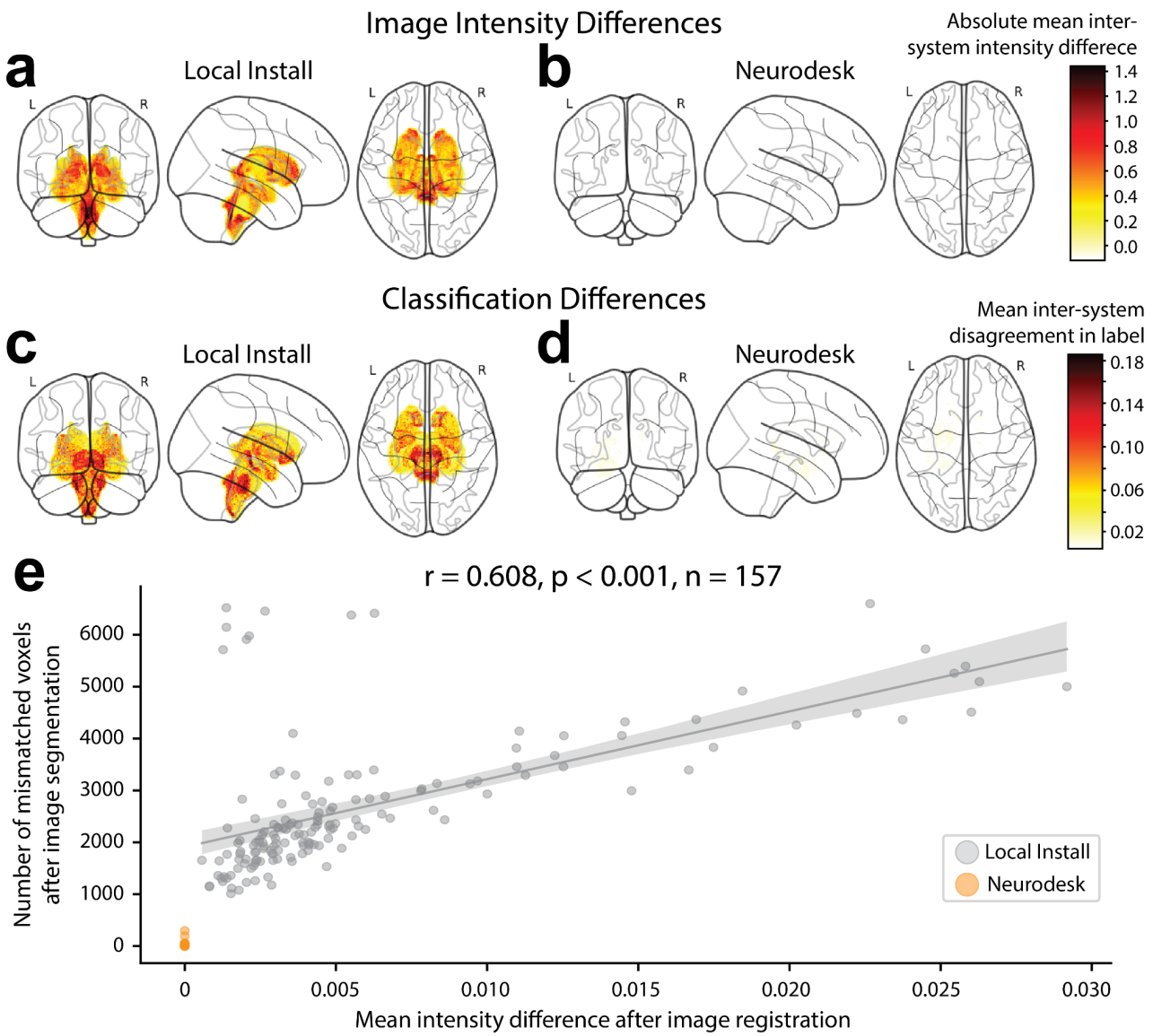


Figure 3. Inter-system differences in image intensity in subcortical structures and subsequent classification of these subcortical structures. (**a,b**) Absolute voxel-wise differences in image intensity within subcortical structures after image registration with FSL-FLIRT on each system (i.e. $|\text{Intensity}_{\text{system A}} - \text{Intensity}_{\text{system B}}|$), averaged across participants. Projections are shown for image registration performed (**a**) using locally installed software and (**b**) using Neurodesk (for which there were no intersystem differences). (**c,d**) Inter-system disagreement in subcortical structure labels after image segmentation with FSL-FIRST, averaged across participants. Projections are shown for image segmentation performed (**c**) using locally installed software and (**d**) using Neurodesk. (**e**) Scatter plot showing, for each participant, the mean inter-system image intensity differences across all voxels within the classified subcortical structures vs. the number of voxels subsequently classified with different labels across systems. For analyses performed with locally installed software, participants with larger differences in image intensity typically also had more prolific disagreement in labels between systems (Pearson's $r = 0.61$, $p < 0.001$). This trend could not be assessed for neurodesk, as there were no differences in image intensity across systems.

Understanding inter-system differences in image registration and tissue classification. Differences in tissue classification were at least partially attributable to differences in registered image intensity earlier in the pipeline. Indeed, there was a strong positive correlation between the magnitude of each participant's inter-system differences in registered image intensity and inter-system classification mismatches (Pearson's $r = 0.608$, $p < .001$, **Figure 3e**). Thus, larger inter-system differences after the FSL FLIRT analysis were associated with larger inter-system differences after the subsequent FSL FIRST analysis.

We next replicated Glatard et al.'s findings by showing that the remaining variability in inter-system differences for tissue-classification, as well as the differences for image registration, could be attributed to a combination of differences in floating point representation and differences in underlying dependencies across systems. Tracing the calls to dynamically linked system libraries revealed many differences for the local installations, but complete congruence between Neurodesk installations (**Figure 4**, see online methods). This begs the question - why were there still minor differences in the classification of subcortical structures for Neurodesk? The most likely explanation is that floating point calculations can produce different results on different processors due to different implementations of the floating point arithmetic instructions[101]. One source of these differences is the presence or absence of fused multiply-add (FMA) instructions, which allow a processor to perform a floating point multiplication and addition in a single instruction. FMA instructions can improve the accuracy and performance of floating point calculations, but their use can also lead to differences in the results of those calculations on different processors, even if they use the same version of the shared library and compiler. Critically, these differences are very small, which is likely why the differences in classification across systems for Neurodesk were so subtle.

Overall, these results demonstrate that differences in dependencies across computing environments can lead to subtle differences in the outcomes of computational analyses, which can snowball across successive processing steps to cause potentially meaningful differences in results across computing environments, especially when investigating subtle effects. By minimising differences at each stage of the analysis, we can enhance the accuracy and reliability of the overall analysis. Critically, Neurodesk eliminates this source of variability by facilitating access to containerised software; thus allowing researchers to reproduce the same result from different computing environments.

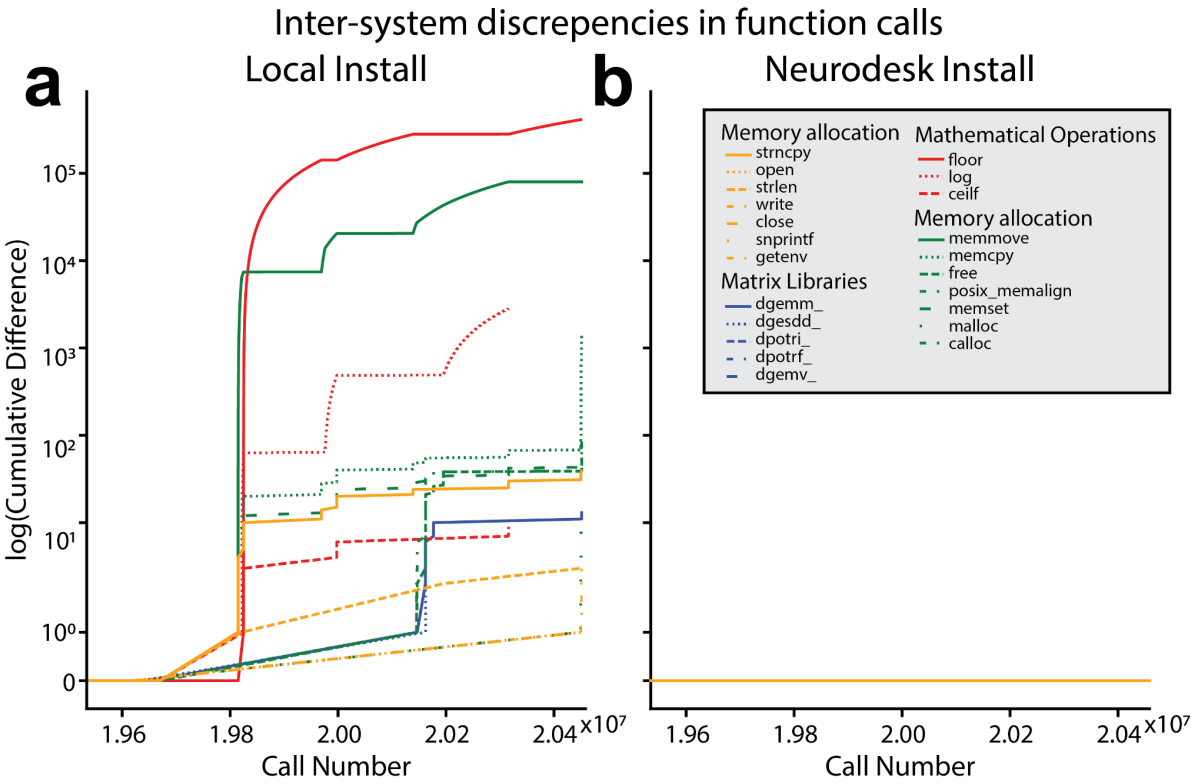


Figure 4. Cumulative difference in the numbers of system library calls between System A and System B for the analysis run using the (**a**) locally installed and (**b**) Neurodesk version of FSL FIRST. Note that calls to *floorf()* were excluded from the plot as they occur earlier in time and the discrepancies for *floorf()* far outnumbered those for any other function from the locally installed tool.

Discussion

Neuroimaging data analysis pipelines are often challenged with limitations in *accessibility*, *portability*, *flexibility*, and *reproducibility*. Neuroscientists may hold back from exploring new tools and/or spend excessive amounts of time installing software (and dependencies) in new computing environments, only to find that the same analysis pipeline produces different results. We developed Neurodesk to address these challenges by building an open-source and community-driven platform for reproducible neuroimaging data analysis. By containerising neuroimaging software, Neurodesk allows scientists to flexibly create fully reproducible and accessible data analysis pipelines which can be run in various computing environments without depreciating over time. By providing an accessibility layer for software containers, the Neurodesk platform allows for convenient portability across computing environments without local software installations. Finally, by keeping the platform open-source and utilising continuous integration and deployment, we have democratised the Neurodesk platform and set a path toward a sustainable ecosystem for neuroimaging data analysis.

The Neurodesk platform has the potential to revolutionise neuroimaging data analysis, not only because it allows for truly reproducible data analysis, but because of how accessible it makes this process. As a group, scientists strive to uphold the scientific principles to the highest possible standard. However, looming deadlines and the pressure to publish often force individual researchers to find a balance between these ideals and the practical constraints imposed by resource limitations. Neurodesk can allow all researchers to adhere to the highest possible standards of reproducibility with minimal changes to their typical development pipelines. Neurodesk enables researchers to not only access a comprehensive suite of neuroimaging data analysis software, but also contribute developments into the future for an ever-increasing suite of packages. Hence, researchers can flexibly take advantage of open datasets, reproduce reported analyses, switch between neuroimaging modalities across projects, and apply complementary analysis methods alongside their primary approach. By harnessing Neurodesk together with cloud computing technologies, published manuscripts can also include links to Jupyter notebooks, democratising reproducibility of key analyses. The ease with which Neurodesk allows analysis pipelines to be shared and reproduced across computing environments also has particular relevance for distributed research groups and collaborative, multi-site projects.

Thus, the Neurodesk platform not only facilitates access to reproducible neuroimaging data analysis, but also makes developing and sharing these workflows less burdensome.

Neurodesk is not the first platform to recognise and seek to address the limited accessibility and reproducibility available for many neuroimaging data analysis tools. Indeed, software distribution mechanisms like Neurodebian[2] have made great progress in making neuroimaging software more accessible, while platforms such as OpenNeuro[33], Brainlife[34], Flywheel[35], XNAT[36] and Qmenta[37] have greatly improved the accessibility and reproducibility of Neuroimaging analyses. However, to date, all existing solutions have lacked portability and flexibility. Many existing solutions require users to upload datasets to their platforms, and developing custom pipelines on these platforms requires substantial platform-specific knowledge. However, even users already accustomed to these specifics may still benefit from the Neurodesk project as Neurodesk's containers are interoperable with other platforms.

Neurodesk has primarily been developed as a research tool to facilitate the analysis of neuroimaging data. However the platform may have a significant impact as an educational tool for workshops, summer schools, and 'hackathons'[102]. The Neurodesk platform was first conceptualised during a 'hackathon' event, during which neuroscientists from around the globe gathered in hubs to collaborate on short-term projects, attend workshops, and develop critical research skills. One of the greatest hurdles for organisers and attendees of such events is the diversity in computing environments across researchers. When delivering a workshop or tutorial, facilitators often spend a large portion of the allocated time troubleshooting installations or issues specific to unique computing environments. Neurodesk addresses these issues by allowing broad access to identical computing environments with the requisite tools pre-installed. This functionality allows groups of researchers to efficiently tackle complex problems by eliminating Sisyphean troubleshooting. The Galaxy platform, for example, has made a significant impact in this way by providing a containerised solution for bioinformatics and social science[103]. Aside from educational applications, Neurodesk can also aid research software developers wishing to make their tools more accessible and efficient to support. The effort to containerise and add one's software to Neurodesk may be minimal compared to the burden of testing across multiple computing platforms and fielding support queries from end-users running software in diverse environments.

Although Neurodesk is already widely used in the community, there are still some potential limitations that warrant discussion. The first is that many of the software containers in the Neurodesk platform currently do not support the ARM CPU architecture, which will become increasingly common in coming years as Mac users update their hardware. This stems from limitations in the underlying software applications, which currently lack support for this processor architecture. However, tool developers are rapidly adapting tools for this architecture and we are convinced that this problem will be addressed for the most commonly used applications in the near future. Further limitations may arise as Neurodesk is applied across more diverse use-cases by the broader research community. A pertinent example relates to the use of proprietary and licensed software. This is an area of active development as the Neurodesk community investigates how such software could be integrated without compromising the accessibility principle. A strength of Neurodesk is that the community-driven, continuous integration model provides a powerful and flexible way to address such expanded use-cases without depending on a single development team. Indeed, this relates to a potential limitation of any such platform: the long term sustainability of the project. The Neurodesk platform was funded with the goal to be sustainable and supported by the community, but for this to be successful the project needs constant maintenance. We therefore developed multiple pathways for sustainability, including the federated support of the underlying hosting infrastructure, flexibility in the continuous integration and deployment infrastructure, and a potential for a commercial model to offer tailored support for institutions and workshops.

The challenges to accessibility and reproducibility posed by neuroimaging data analysis software are not unique to neuroscience. While we have chosen to containerise software designed for neuroimaging datasets, the principles governing the design of the Neurodesk platform need not be restricted to this field of research. This open-source platform could be used to deploy software specific to any other discipline, and it is our sincere hope that this platform is adapted to other disciplines struggling with similar issues. The Neurodesk platform has the potential to profoundly improve the way scientists analyse their data and communicate their results. For the first time, this platform allows any scientist, anywhere in the world, to conveniently access their data analysis tools and apply these tools in a fully reproducible manner from any computing environment. We are excited to see what new insights such technology can enable.

Online methods

Neurodesk's open-access code and documentation

All stages of development from the initial conception as a hackathon project, through to the most current iteration of Neurodesk, with up-to-date community-built Neurocontainer recipes, are documented publicly:

https://www.neurodesk.org/ - Platform website which includes 'Getting Started' tutorials for new users of various skill-levels

https://github.com/NeuroDesk - Public GitHub repository, where Issues can be logged, and contributions can be made by any community member with a GitHub account and the eagerness to create pull requests.

Frequently Asked Questions

Will running my analyses on Neurodesk be slower than if they were run locally, especially if I'm on a slower internet connection?

The internet bandwidth will only affect the speed of your analysis the first time you use a new tool. Neurodesk uses the Cern Virtual File Management system (CVMFS), which means that only the specific part of a container which is currently used will be downloaded over the internet. Once downloaded, these will be cached locally, meaning that software will operate at the same speed as it would when running locally (see **table S1**). Although there is a container initialization time that could impact performance in comparison to a non-containerized workflow, there is evidence that in some cases containerised analysis pipelines may run even faster than locally installed software due to efficiency gains in accessing files[104] .

Where are Neurodesk containers stored, and will the performance differ from country to country?

Neurodesk containers are distributed globally via CVMFS and accessed from the fastest server according to your location. Our goal is to get mirror servers as close as

possible to all users, so that CVMFS can automatically switch to another mirror server if one fails.

Are there any security concerns regarding using the Neurodesk platform in a web browser? For example, could there be any risks that compromise data processed on Neurodesk?

The underlying container technology in Neurodesk ensures that applications are isolated with least privileges, to minimise the impact of malware attacks. Interacting with the web from within a Neurodesktop poses a similar risk to any system with access to the internet, so all similar precautions would apply. Neurodesktops can be shut down, deleted and started fresh with minimal effort, which means recovery is significantly simpler than a native installation in a similar scenario. To ensure data security, it is important for users who run Neurodesk on a cloud provider to follow security best practices. For an in depth review of the potential security concerns involved in containerising scientific data analysis software, see Kaur et al (2021)[105].

Can I store processed data in Neurodesk?

Neurodesktop allows host directories to be mounted for internal data access, and these directories can then be accessed from the Neurocontainers. Data can also be accessed via data access clients and the web inside a Neurodesktop instance.

Can you provide more technical detail on how the Neurodesk desktop virtual environment has been built?

Neurodesktop is a Docker container packaging a linux desktop environment that delivers neuroscience applications via CVMFS, wrapped up as singularity containers. It uses Apache Guacamole with underlying remote-desktop protocol (RDP) or virtual network computing (VNC) remote desktop protocols to deliver a desktop experience in the browser, with copy, paste and file transfer functionality.

Why are there different types of containers (i.e. Docker, Singularity) in Neurodesktop? Are there any conflicts between Docker and Singularity?

Docker and Singularity containers are both used in Neurodesktop for different, complementary purposes. Docker is used to containerise the Neurodesktop environment due its cross-platform support and ability to run singularity containers within. Singularity,

which is used for the individual application containers (Neurocontainers) is preferred by most high performance computing (HPC) platforms, where multi-user security and scheduling are of particular concern) and can also be used indirectly via wrapper scripts and lmod; a system which manages environment configurations for different software packages.

Are there any financial costs associated with keeping Neurodesk running, and if so, how will these be met for the foreseeable future?

The long term sustainability of Neurodesk has been planned according to three possible financial scenarios. *1) No further funding*: In this case, Neurodesk will be minimally maintained such that all the open access containers will still be accessible. However, Neurodesk Play (the cloud-based no-install version of Neurodesktop) will no longer be accessible and the software distribution via CVMFS Neurodesk may run more slowly outside of Australia. *2) Marginal Funding.* Neurodesk will be maintained with its current functionality, but with less focus on development of new features. *3) Sufficient funding*. The Neurodesk team is working on a not-for-profit business model in which additional financial costs involved in increasing Neurodesk's current functionality could be covered by charging a nominal fee to manage the resources required to deploy Neurodesk in the cloud for organisations or for workshop and teaching purposes. Note that Neurodesk (Neurodesktop, Neurocommand, and the Neurocontainers) will always remain open-source and open-access under the MIT licence, which enables commercial use. Any fee would be used to reduce the administrative load and technical challenges for workshop organisers and participants, such that workshop participants can access a fully maintained and cloud-based Neurodesktop environment.

Neurodesk is open-source, such that anyone is able to contribute containerised software to the platform. Are there any protocols in place to verify that this software is working as expected before it is made available to the community?

There is a feature to include a functional test within each tool's container. This test can be run automatically after each container is built. However, such automated tests can only cover a subset of potential problems and we also rely on issues reported by users on GitHub and manual testing of new containers when releasing new versions.

The software I need is not available in Neurodesk, and I don't feel confident in my ability to contribute a container to the Neurodesk repository. Is there a way I can request that it be added?

Users can submit a GitHub issue to request new tools by providing the following information: name and version of the tool, preferred Linux distribution, Linux commands for installing the tool, any GUI applications and commands to start them, test data to include, reference to paper, link to documentation, and commands for testing the tool.

How do I get help if I encounter an issue with Neurodesk?

There is an active discussion forum on GitHub with a Q&A section. If your question has not already been addressed there, please raise a new issue.

Reproducibility in Neurodesk

To investigate our claims that the Neurodesk platform's containerised tools lead to more reproducible results than locally installed software, we sought to conceptually replicate the results reported by Glatard et al. (2015) using Neurodesk vs locally installed software across different operating systems. The first steps in Glatard et al.'s analysis pipeline were brain extraction and tissue classification.

Brain extraction and tissue classification. We began by running FSL BET and FAST on raw MRI images to extract voxels containing brain tissue and classify tissue types, respectively. The file checksums for the outputs of these processing steps were identical across all computing environments, verifying that the implementation of the processing pipeline was reproducible across systems for both Neurodesk and local installation. After these steps, we performed image registration and tissue classification with FSL-FLIRT and FSL-FIRST, respectively. These analysis steps did lead to differences in results across systems, and are thus reported in the main text.

Understanding inter-system differences in image registration and tissue classification. Given that the image registration and tissue classification steps led to inter-system differences, we sought to understand the cause of these differences. FSL utilises dynamic linking to shared system libraries such as libmath and LAPACK, which are loaded at runtime. Thus, while the same version of FSL was installed in all four computing environments, differences in image processing still emerge for analyses run on locally installed software. This is due to

differences in dependencies across systems, a problem circumvented by Neurodesk. To better understand how such differences might emerge, calls to these libraries were recorded for a representative image using 'ltrace'. The libraries called during the FLIRT and FIRST analyses could be categorised into four main classes: mathematical operations, matrix operations, memory allocation, and system operations. Interestingly, Glatard et al., who used older software versions than we investigated here, found that image processing differences across systems resulted largely from differences in floating point representation in the mathematical functions *expf()*, *cosf()*, and *sinf()*. They also found inter-system differences in the control-flow of the programs, indicated by differences in the number of library calls to mathematical functions such as *floorf()*. Here, differences in floating point representation were less severe, as these were only present for the *sinf()* function. However, the number of calls made to several functions differed across the local FSL installations, indicating that the inter-system differences in the control flow of the processing pipeline remain an issue for reproducibility (**Table S1**). The *floorf()* function represented the most prevalent difference in library calls. There were over 13 thousand additional calls to this function made on System B relative to System A for the FLIRT analysis, and approximately 5.5 million additional calls for the FIRST analysis. Overall, the FIRST analysis had greater discrepancy in calls overall. After accounting for the additional calls to *floorf()*, which occurred early in the FIRST analysis pipeline, mismatches in the sequence of system calls to several other functions remained (**Figure 4a**). However, all remaining mismatches across systems occurred in memory allocation functions. Importantly, there was no difference in floating point representation or the number of system calls to shared libraries across systems for the Neurodesk implementation of FSL (**Figure 4b**), while maintaining a similar runtime as local installation on the same hardware (**Table S1**).

Table S1. Differences in execution of tissue segmentation (FIRST) and image registration (FLIRT) pipelines.

	Local		**Neurodesk**	
FIRST (# of calls)	**System A**	**System B**	**System A**	**System B**
floor	553,308	553,962	553,341	553,341
floorf	48,406,500	53,942,784	51,928,356	51,928,356
log	2,820	3,138	3,024	3,024
FLIRT (# of calls)	**System A**	**System B**	**System A**	**System B**
floorf	41,347,920	41,334,549	41,342,544	41,342,544
Runtime (n=9)	**System A**	**System B**	**System A**	**System B**
Average (mins)	1.57	3.89	1.69	3.59
Standard Deviation (mins)	0.03	0.13	0.04	0.11

Understanding the practical implications of inter-system differences. The local installations led to inter-system differences in tissue classification orders of magnitude larger than Neurodesk. However, it is difficult to know how voxel-wise differences of this scale might actually affect test statistics i.e. could I actually come to a different conclusion about my research question if I ran the same analysis on the same data on a different computer? To address these questions, we performed a permutation test to examine the impact of inter-system differences in tissue classification (using FSL FIRST) on correlations between subcortical structure volumes and age.

On each system (A,B), for both Neurodesk and local installations, we computed the volume of each subcortical structure in the left hemisphere, right hemisphere, and the whole structure by participant. We performed Monte Carlo permutation tests for each of these volumes (9999 permutations each). On each permutation, we performed a Pearson

correlation of volume vs. participant age, and calculated the differences in the values of the correlation coefficient's across the two systems. These permutation tests were repeated for three different sample sizes (n=10, 30, 50), such that each permutation for each sample-size represented a different randomly selected group of participants. Critically, for each sample-wise permutation, the same sample was used for each of the two systems, such that the test-statistic difference always represented inter-system differences rather than inter-sample differences. Thus, the distribution of test statistic differences for each sample size represents 219978 permuted samples (8 subcortical structures (Putamen, Amygdala, Thalamus, Pallidum, Caudate Nucleus, Hippocampus, Brain-Stem, Accumbens.) x 3 methods (left hemisphere, right hemisphere, both) x 9999 subject-wise permutations).

The analysis showed that as sample size decreased, the inter-system coefficient differences for the local installations increased in magnitude (Local installation: N=50, Δr = -0.02 – 0.02 | N=30, Δr = -0.04 – 0.03 | N=10, Δr = -0.08 – 0.11; **Figure S1**). By contrast, the inter-system test statistic differences for Neurodesk were negligible, and did not scale with sample size (Neurodesk: N=50, $\Delta r = -1.74 \times 10^{-3} - 2.59 \times 10^{-4}$ | N=30, $\Delta r = -3.75 \times 10^{-5} - 1.89 \times 10^{-4}$ | N=10, $\Delta r = -1.52 \times 10^{-3} - 0$; **Figure S1**). Thus, the minor differences in image processing with locally installed software can meaningfully impact the reliability of test statistics, especially when statistical power is already low. It is therefore crucial to consider both sample variability and system when conducting these types of analyses.

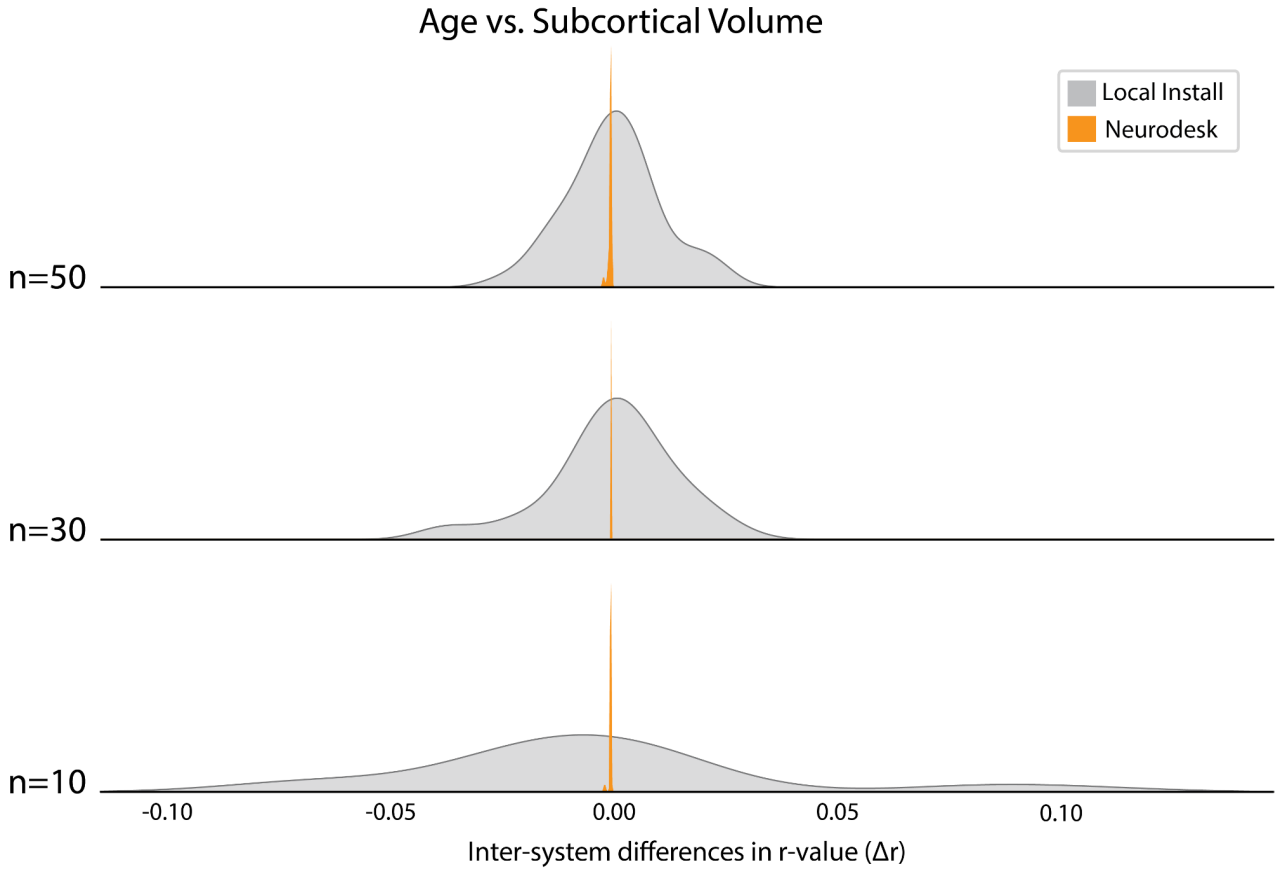


Figure S1. Permutation test results showing inter-system differences in r-values for the correlation between age and volume of subcortical structures, organised by sample-size (n = 10, 30, 50).

References

1. Brand, A., Allen, L., Altman, M., Hlava, M. & Scott, J. Beyond authorship: attribution, contribution, collaboration, and credit. *Learn. Publ.* **28**, 151–155 (2015).
2. Halchenko, Y. & Hanke, M. Open is Not Enough. Let's Take the Next Step: An Integrated, Community-Driven Computing Platform for Neuroscience. *Front. Neuroinformatics* **6**, 22 (2012).
3. Hanke, M. & Halchenko, Y. Neuroscience Runs on GNU/Linux. *Front. Neuroinformatics* **5**, 8 (2011).
4. Niso, G. *et al.* Open and reproducible neuroimaging: From study inception to publication. *NeuroImage* **263**, 119623 (2022).
5. The FAIR Guiding Principles for scientific data management and stewardship | Scientific Data. https://www.nature.com/articles/sdata201618.
6. Kurtzer, G. M., Sochat, V. & Bauer, M. W. Singularity: Scientific containers for mobility of compute. *PLOS ONE* **12**, e0177459 (2017).
7. Amari, S.-I. *et al.* Neuroinformatics: the integration of shared databases and tools towards integrative neuroscience. *J. Integr. Neurosci.* **01**, 117–128 (2002).
8. Van Gorp, P. & Mazanek, S. SHARE: a web portal for creating and sharing executable research papers. *Procedia Comput. Sci.* **4**, 589–597 (2011).
9. Berg, J. Progress on reproducibility. *Science* **359**, 9–9 (2018).

10. Begley, C. G. & Ioannidis, J. P. A. Reproducibility in Science. *Circ. Res.* **116**, 116–126 (2015).

11. Goodman, S. N., Fanelli, D. & Ioannidis, J. P. A. What does research reproducibility mean? *Sci. Transl. Med.* **8**, 341ps12-341ps12 (2016).

12. Johnson, V. E., Payne, R. D., Wang, T., Asher, A. & Mandal, S. On the Reproducibility of Psychological Science. *J. Am. Stat. Assoc.* **112**, 1–10 (2017).

13. Nosek, B. A. *et al.* Replicability, Robustness, and Reproducibility in Psychological Science. Preprint at https://doi.org/10.31234/osf.io/ksfvq (2021).

14. OPEN SCIENCE COLLABORATION. Estimating the reproducibility of psychological science. *Science* **349**, aac4716 (2015).

15. Claerbout, J. F. & Karrenbach, M. Electronic documents give reproducible research a new meaning. in *SEG technical program expanded abstracts 1992* 601–604 (Society of Exploration Geophysicists, 1992).

16. Cacioppo, J. T., Kaplan, R. M., Krosnick, J. A., Olds, J. L. & Dean, H. Social, behavioral, and economic sciences perspectives on robust and reliable science. *Rep. Subcomm. Replicability Sci. Advis. Comm. Natl. Sci. Found. Dir. Soc. Behav. Econ. Sci.* (2015).

17. Nosek, B. A. *et al.* Promoting an open research culture. *Science* **348**, 1422–1425 (2015).

18. Glatard, T. *et al.* Reproducibility of neuroimaging analyses across operating systems. *Front. Neuroinformatics* **9**, (2015).

19. Schöbi, D. *et al.* Technical note: A fast and robust integrator of delay differential equations in DCM for electrophysiological data. *NeuroImage* **244**, 118567 (2021).

20. Frässle, S. *et al.* Test-retest reliability of dynamic causal modeling for fMRI. *NeuroImage* **117**, 56–66 (2015).

21. Li, X. *et al.* Evaluating and Improving Cross-Pipeline Reproducibility in Functional Connectomics: A Case Study.

22. Silberzahn, R. *et al. Many analysts, one dataset: Making transparent how variations in analytical choices affect results. Advances in Methods and Practices in Psychological Science* (2017).

23. Abe, T. *et al.* Neuroscience Cloud Analysis As a Service. 2020.06.11.146746 Preprint at https://doi.org/10.1101/2020.06.11.146746 (2021).

24. Tapera, T. M. *et al.* FlywheelTools: Data Curation and Manipulation on the Flywheel Platform. *Front. Neuroinformatics* **15**, (2021).

25. Routier, A. *et al.* Clinica: An Open-Source Software Platform for Reproducible Clinical Neuroscience Studies. *Front. Neuroinformatics* **15**, 689675 (2021).

26. NeuroFedora: Free software for Free neuroscience :: Fedora Docs. https://docs.fedoraproject.org/en-US/neurofedora/overview/.

27. Boettiger, C. An introduction to Docker for reproducible research. *ACM SIGOPS Oper. Syst. Rev.* **49**, 71–79 (2015).

28. Rad, B. B., Bhatti, H. J. & Ahmadi, M. An introduction to docker and analysis of its performance. *Int. J. Comput. Sci. Netw. Secur. IJCSNS* **17**, 228 (2017).

29. Trunov, A. S., Voronova, L. I., Voronov, V. I. & Ayrapetov, D. P. Container Cluster Model Development for Legacy Applications Integration in Scientific Software System. in *2018 IEEE International Conference 'Quality Management, Transport and Information Security, Information Technologies' (IT QM IS)* 815–819 (2018). doi:10.1109/ITMQIS.2018.8525120.

30. BioContainers: an open-source and community-driven framework for software standardization | Bioinformatics | Oxford Academic. https://academic.oup.com/bioinformatics/article/33/16/2580/3096437?login=true.

31. Wagner, A. S. *et al.* FAIRly big: A framework for computationally reproducible processing of large-scale data. *Sci. Data* **9**, 80 (2022).

32. Avesani, P. *et al.* The open diffusion data derivatives, brain data upcycling via integrated publishing of derivatives and reproducible open cloud services. *Sci. Data* **6**, 69 (2019).

33. Markiewicz, C. J. *et al.* The OpenNeuro resource for sharing of neuroscience data. *eLife* **10**, e71774 (2021).

34. brainlife. https://brainlife.io/about/.

35. Flywheel | Biomedical Research Data Platform. *Flywheel* https://flywheel.io/.

36. Herrick, R. *et al.* XNAT Central: Open sourcing imaging research data. *NeuroImage* **124**, 1093–1096 (2016).

37. The all in one platform for your imaging clinical trial or study. https://www.qmenta.com.

38. Blomer, J. *et al.* Micro-CernVM: slashing the cost of building and deploying virtual machines. *J. Phys. Conf. Ser.* **513**, 032009 (2014).

39. Gorgolewski, K. *et al.* Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python. *Front. Neuroinformatics* **5**, (2011).

40. Adebimpe, A. *et al.* ASLPrep: a platform for processing of arterial spin labeled MRI and quantification of regional brain perfusion. *Nat. Methods* **19**, 683–686 (2022).

41. Esteban, O. *et al.* fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat. Methods* **16**, 111–116 (2019).

42. Esteban, O. *et al.* MRIQC: Advancing the automatic prediction of image quality in MRI from unseen sites. *PLOS ONE* **12**, e0184661 (2017).

43. Stewart, A. W. *et al.* QSMxT: Robust masking and artifact reduction for quantitative susceptibility mapping. *Magn. Reson. Med.* **87**, 1289–1300 (2022).

44. Zwiers, M. P., Moia, S. & Oostenveld, R. BIDScoin: A User-Friendly Application to Convert Source Data to Brain Imaging Data Structure. *Front. Neuroinformatics* **15**, (2022).

45. Gorgolewski, K. J. *et al.* The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci. Data* **3**, 160044 (2016).

46. Yushkevich, P. A. *et al.* User-Guided Segmentation of Multi-modality Medical Imaging Datasets with ITK-SNAP. *Neuroinformatics* **17**, 83–102 (2019).

47. Wang, R., Benner, T., Sorensen, A. G. & Wedeen, V. J. Diffusion Toolkit: A Software Package for Diffusion Imaging Data Processing and Tractography.

48. Diffusion MRI Reconstruction in DSI Studio - DSI Studio—A Tractography Software Tool. https://sites.google.com/a/labsolver.org/dsi-studio/Manual/Reconstruction?overridemobile=true.

49. Tournier, J.-D., Calamante, F. & Connelly, A. MRtrix: diffusion tractography in crossing fiber regions. *Int. J. Imaging Syst. Technol.* **22**, 53–66 (2012).

50. Dhollander, T. MRtrix3Tissue. *MRtrix3Tissue* https://3tissue.github.io/.

51. Pallast, N. *et al.* Processing Pipeline for Atlas-Based Imaging Data Analysis of Structural and Functional Mouse Brain MRI (AIDAmri). *Front. Neuroinformatics* **13**, (2019).

52. Rodent Automated Bold Improvement of EPI Sequences (RABIES): A standardized image processing and data quality platform for rodent fMRI | bioRxiv. https://www.biorxiv.org/content/10.1101/2022.08.20.504597v1.abstract.

53. LCModel's home page. http://s-provencher.com/lcmodel.shtml.

54. Hangel, G. *et al.* Ultra-high resolution brain metabolite mapping at 7 T by short-TR Hadamard-encoded FID-MRSI. *NeuroImage* **168**, 199–210 (2018).

55. Cox, R. W. AFNI: What a long strange trip it's been. *NeuroImage* **62**, 743–747 (2012).

56. Avants, B. B., Tustison, N. & Johnson, H. Advanced Normalization Tools (ANTS).

57. Wisse, L. E. M. *et al.* Automated Hippocampal Subfield Segmentation at 7T MRI. *Am. J. Neuroradiol.* **37**, 1050–1057 (2016).

58. BART Toolbox. https://mrirecon.github.io/bart/.

59. Gaser, C. *et al.* CAT – A Computational Anatomy Toolbox for the Analysis of Structural MRI Data. 2022.06.11.495736 Preprint at https://doi.org/10.1101/2022.06.11.495736 (2022).

60. Eckstein, K. Susceptibility Weighted Imaging (CLEAR-SWI). (2022).

61. Whitfield-Gabrieli, S. & Nieto-Castanon, A. Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. *Brain Connect.* **2**, 125–141 (2012).

62. Marcus, D. S. *et al.* Human Connectome Project informatics: Quality control, database services, and data visualization. *NeuroImage* **80**, 202–219 (2013).

63. Estrada, S. *et al.* FatSegNet: A fully automated deep learning pipeline for adipose tissue segmentation on abdominal dixon MRI. *Magn. Reson. Med.* **83**, 1471–1483 (2020).

64. Fischl, B. FreeSurfer. *NeuroImage* **62**, 774–781 (2012).

65. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M. FSL. *NeuroImage* **62**, 782–790 (2012).

66. Isensee, F. *et al.* Automated brain extraction of multisequence MRI using artificial neural networks. *Hum. Brain Mapp.* **40**, 4952–4964 (2019).

67. Shaw, T., York, A., Ziaei, M., Barth, M. & Bollmann, S. Longitudinal Automatic Segmentation of Hippocampal Subfields (LASHiS) using multi-contrast MRI.

NeuroImage **218**, 116798 (2020).

68. Huber, L. R. *et al.* LayNii: A software suite for layer-fMRI. *NeuroImage* **237**, 118091 (2021).
69. Vincent, R. D. *et al.* MINC 2.0: A Flexible Format for Multi-Modal Images. *Front. Neuroinformatics* **10**, (2016).
70. Grussu, F. *et al.* Multi-parametric quantitative in vivo spinal cord MRI with unified signal readout and image denoising. *NeuroImage* **217**, 116884 (2020).
71. NITRC: NiftyReg: Tool/Resource Info. https://www.nitrc.org/projects/niftyreg/.
72. NITRC: NiiStat: Tool/Resource Info. https://www.nitrc.org/projects/niistat/.
73. Chang, J. *et al.* Open-Source Hypothalamic-ForniX (OSHy-X) Atlases and Segmentation Tool for 3T and 7T. *J. Open Source Softw.* **7**, 4368 (2022).
74. Winkler, A. M., Ridgway, G. R., Webster, M. A., Smith, S. M. & Nichols, T. E. Permutation inference for the general linear model. *NeuroImage* **92**, 381–397 (2014).
75. Kasper, L. *et al.* The PhysIO toolbox for modeling physiological noise in fMRI data. *J. Neurosci. Methods* **276**, 56–72 (2017).
76. Dymerska, B. *et al.* Phase unwrapping with a rapid opensource minimum spanning tree algorithm (ROMEO). *Magn. Reson. Med.* **85**, 2294–2308 (2021).
77. Fedorov, A. *et al.* 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. *Magn. Reson. Imaging* **30**, 1323–1341 (2012).

78. De Leener, B. *et al.* SCT: Spinal Cord Toolbox, an open-source software for processing spinal cord MRI data. *NeuroImage* **145**, 24–43 (2017).

79. Ashburner, J. Computational anatomy with the SPM software. *Magn. Reson. Imaging* **27**, 1163–1174 (2009).

80. Langkammer, C. *et al.* Fast quantitative susceptibility mapping using 3D EPI and total generalized variation. *NeuroImage* **111**, 622–630 (2015).

81. Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D. & Leahy, R. M. Brainstorm: A user-friendly application for MEG/EEG analysis. *Comput. Intell. Neurosci.* **2011**, 879716 (2011).

82. Brunner, C., Delorme, A. & Makeig, S. Eeglab – an Open Source Matlab Toolbox for Electrophysiological Research. **58**, (2013).

83. Oostenveld, R., Fries, P., Maris, E. & Schoffelen, J.-M. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Intell Neurosci.* **2011**, 1–9 (2011).

84. Gramfort, A. *et al.* MNE software for processing MEG and EEG data. *Neuroimage* **86**, 446–460 (2014).

85. Brunner, C., Breitwieser, C. & Müller-Putz, G. R. Sigviewer and Signalserver – Open Source Software Projects for Biosignal Analysis. **58**, (2013).

86. Ihaka, R. & Gentleman, R. R: A Language for Data Analysis and Graphics. *J. Comput.*

Graph. Stat. **5**, 299–314 (1996).

87. Ribeiro, F. L., Bollmann, S. & Puckett, A. M. Predicting the retinotopic organization of human visual cortex from anatomy using geometric deep learning. *NeuroImage* **244**, 118624 (2021).
88. Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W. & Popa, R. A. Delphi: A cryptographic inference service for neural networks. in *29th USENIX Security Symposium (USENIX Security 20)* 2505–2522 (2020).
89. Still, M. *The definitive guide to ImageMagick*. vol. 1 (Springer, 2006).
90. GIMP. *GIMP* https://www.gimp.org/.
91. NITRC: MRIcron: Tool/Resource Info. https://www.nitrc.org/projects/mricron.
92. Image Visualization with MRIcroGL — Andy's Brain Book 1.0 documentation. https://andysbrainbook.readthedocs.io/en/latest/MRIcroGL/MRIcroGL_Overview.html.
93. Vicory, J. *et al.* SlicerSALT: Shape AnaLysis Toolbox. in *Shape in Medical Imaging* (eds. Reuter, M. et al.) vol. 11167 65–72 (Springer International Publishing, 2018).
94. Rorden, C. Surf Ice. (2022).
95. Plesser, H. E. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Front. Neuroinformatics* **11**, (2018).
96. Peng, R. D. Reproducible Research in Computational Science. *Science* **334**, 1226–1227 (2011).

97. Donoho, D. L., Maleki, A., Rahman, I. U., Shahram, M. & Stodden, V. Reproducible Research in Computational Harmonic Analysis. *Comput. Sci. Eng.* **11**, 8–18 (2009).

98. Gronenschild, E. H. *et al.* The effects of FreeSurfer version, workstation type, and Macintosh operating system version on anatomical volume and cortical thickness measurements. *PloS One* **7**, e38234 (2012).

99. Krefting, D. *et al.* Reliability of quantitative neuroimage analysis using freesurfer in distributed environments. in *MICCAI Workshop on High-Performance and Distributed Computing for Medical Imaging.(Toronto, ON)* (2011).

100. Mazziotta, J. *et al.* A probabilistic atlas and reference system for the human brain: International Consortium for Brain Mapping (ICBM). *Philos. Trans. R. Soc. Lond. Ser. B* **356**, 1293–1322 (2001).

101. The pitfalls of verifying floating-point computations | ACM Transactions on Programming Languages and Systems. https://dl.acm.org/doi/abs/10.1145/1353445.1353446.

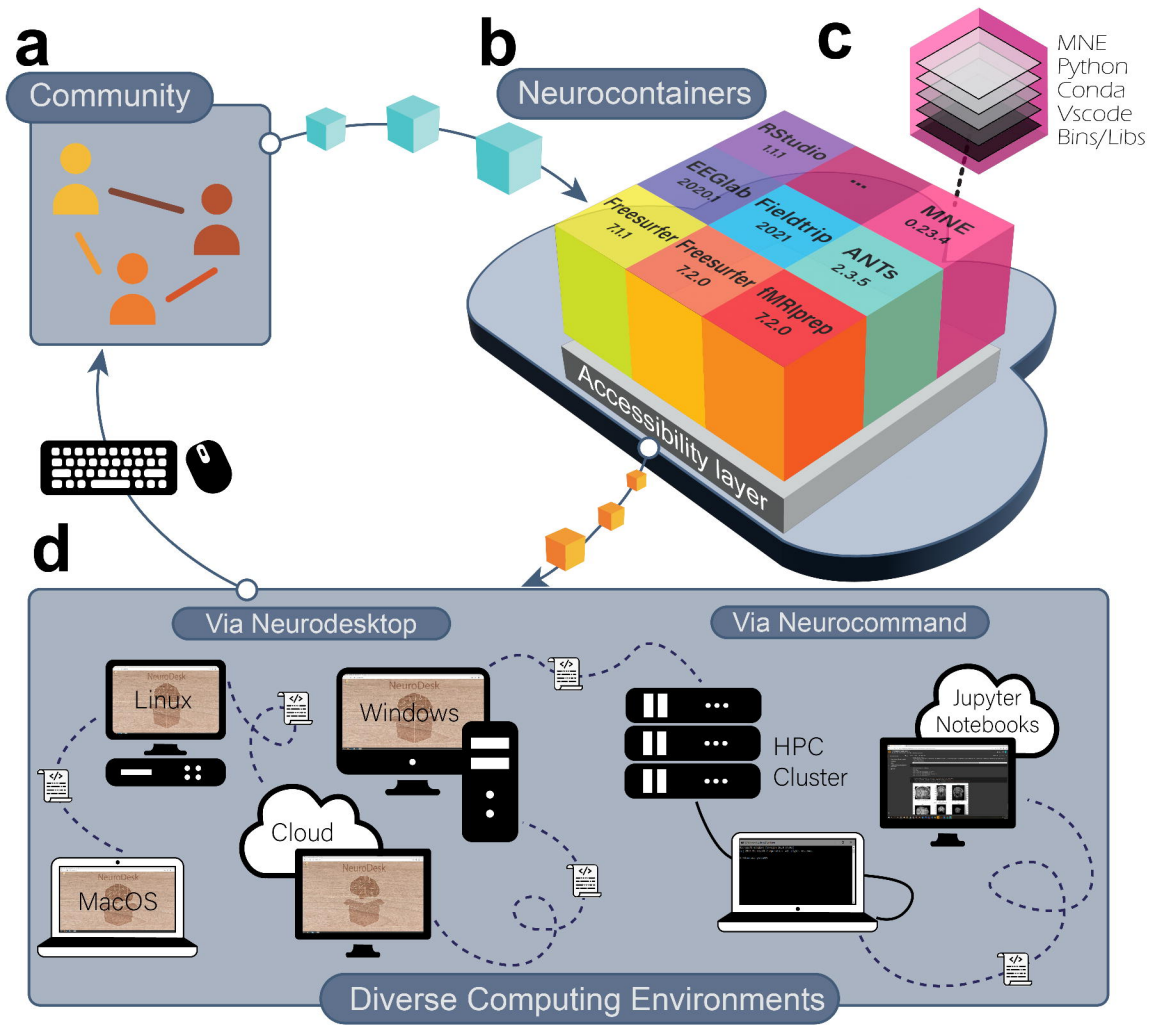
102. Gau, R. *et al.* Brainhack: Developing a culture of open, inclusive, community-driven neuroscience. *Neuron* **109**, 1769–1775 (2021).

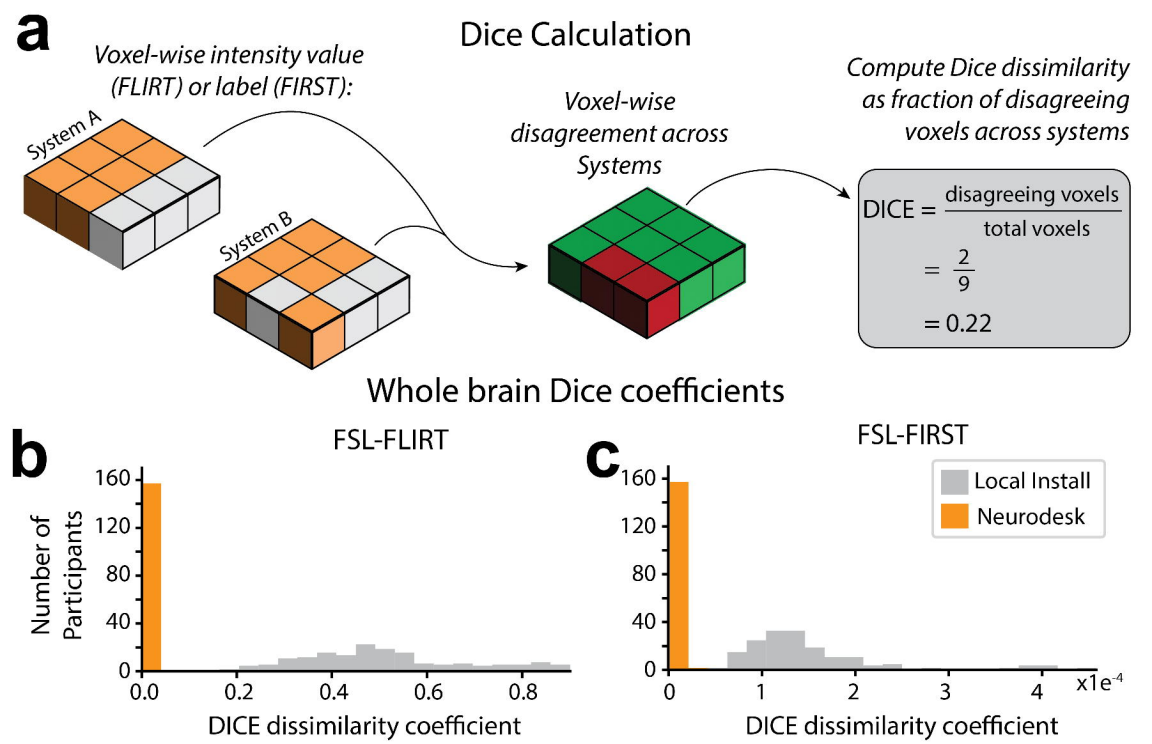
103. Afgan, E. *et al.* The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.* **46**, W537–W544 (2018).

104. Rioux, P., Kiar, G., Hutton, A., Evans, A. C. & Brown, S. T. Deploying large fixed file

datasets with SquashFS and Singularity. Preprint at https://doi.org/10.48550/arXiv.2002.06129 (2020).

105. Kaur, B., Dugré, M., Hanna, A. & Glatard, T. An analysis of security vulnerabilities in container images for scientific data analysis. *GigaScience* **10**, giab025 (2021).


a
Community
b
Neurocontainers
c
MNE
Python
Conda
Vscode
Bins/Libs
RStudio
1.1.1
EEGlab
2020.1
...
Freesurfer
7.1.1
Fieldtrip
2021
MNE
0.23.4
Freesurfer
7.2.0
ANTs
2.3.5
fMRIprep
7.2.0
Accessibility layer
d
Via Neurodesktop
Via Neurocommand
Linux
Windows
Cloud
MacOS
HPC
Cluster
Jupyter
Notebooks
Diverse Computing Environments


a
Dice Calculation
Voxel-wise intensity value (FLIRT) or label (FIRST):
System A
System B
Voxel-wise disagreement across Systems
Compute Dice dissimilarity as fraction of disagreeing voxels across systems
DICE = disagreeing voxels / total voxels
= 2/9
= 0.22
Whole brain Dice coefficients
b
FSL-FLIRT
Number of Participants
160
120
80
40
0
0.0
0.2
0.4
0.6
0.8
DICE dissimilarity coefficient
c
FSL-FIRST
Local Install
Neurodesk
0
1
2
3
4
x1e-4
DICE dissimilarity coefficient

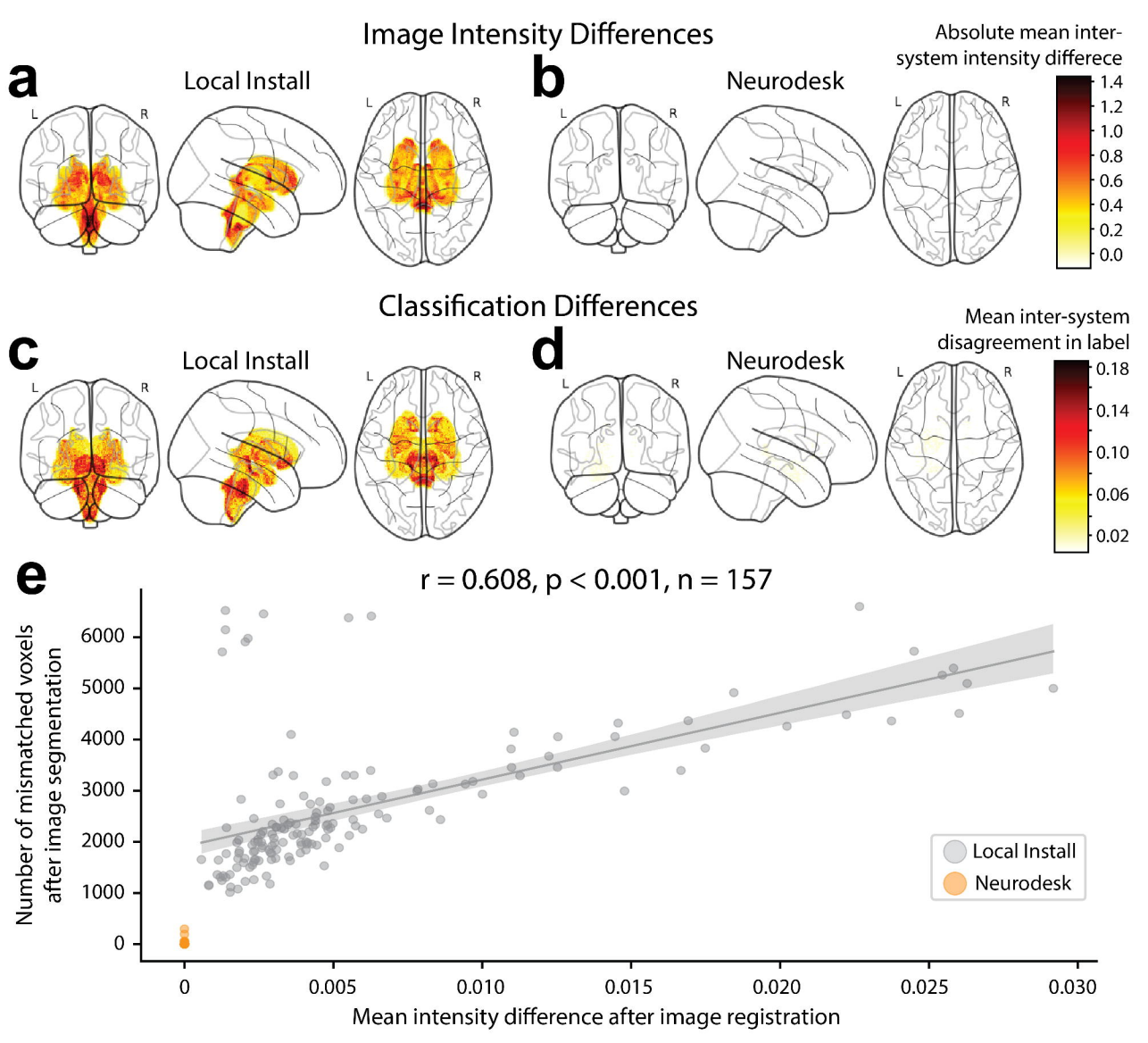
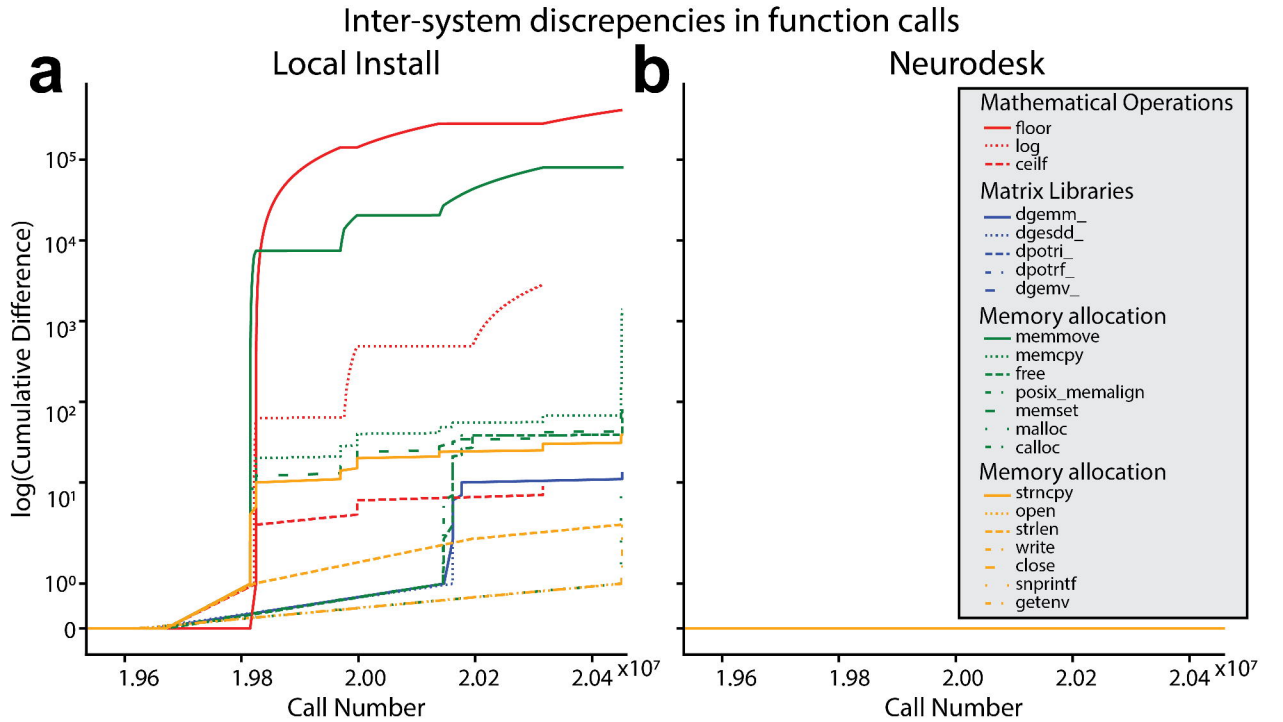
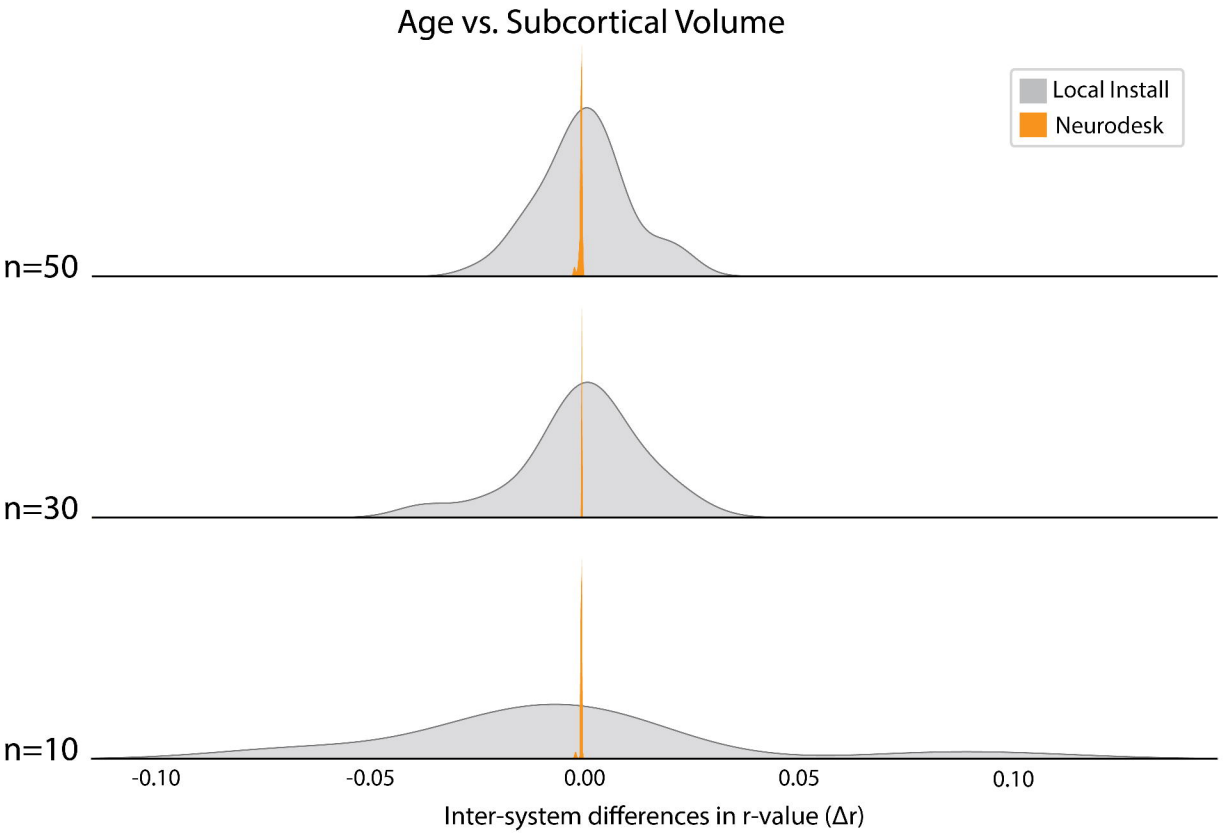


Image Intensity Differences
a
Local Install
b
Neurodesk
Absolute mean inter-system intensity differece
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Classification Differences
c
Local Install
d
Neurodesk
Mean inter-system disagreement in label
0.18
0.14
0.10
0.06
0.02
e
r = 0.608, p < 0.001, n = 157
Number of mismatched voxels after image segmentation
Mean intensity difference after image registration
Local Install
Neurodesk


Inter-system discrepencies in function calls
a
Local Install
b
Neurodesk
log(Cumulative Difference)
10^5
10^4
10^3
10^2
10^1
10^0
0
1.96
1.98
2.00
2.02
2.04
$\times 10^7$
Call Number
Mathematical Operations
floor
log
ceilf
Matrix Libraries
dgemm_
dgesdd_
dpotri_
dpotrf_
dgemv_
Memory allocation
memmove
memcpy
free
posix_memalign
memset
malloc
calloc
Memory allocation
strncpy
open
strlen
write
close
snprintf
getenv


Age vs. Subcortical Volume
Local Install
Neurodesk
n=50
n=30
n=10
-0.10
-0.05
0.00
0.05
0.10
Inter-system differences in r-value (Δr)