nature methods

Brief Communication

https://doi.org/10.1038/s41592-023-02145-x

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

Table of Contents

Reproducibility in Neurodesk: A Case Study

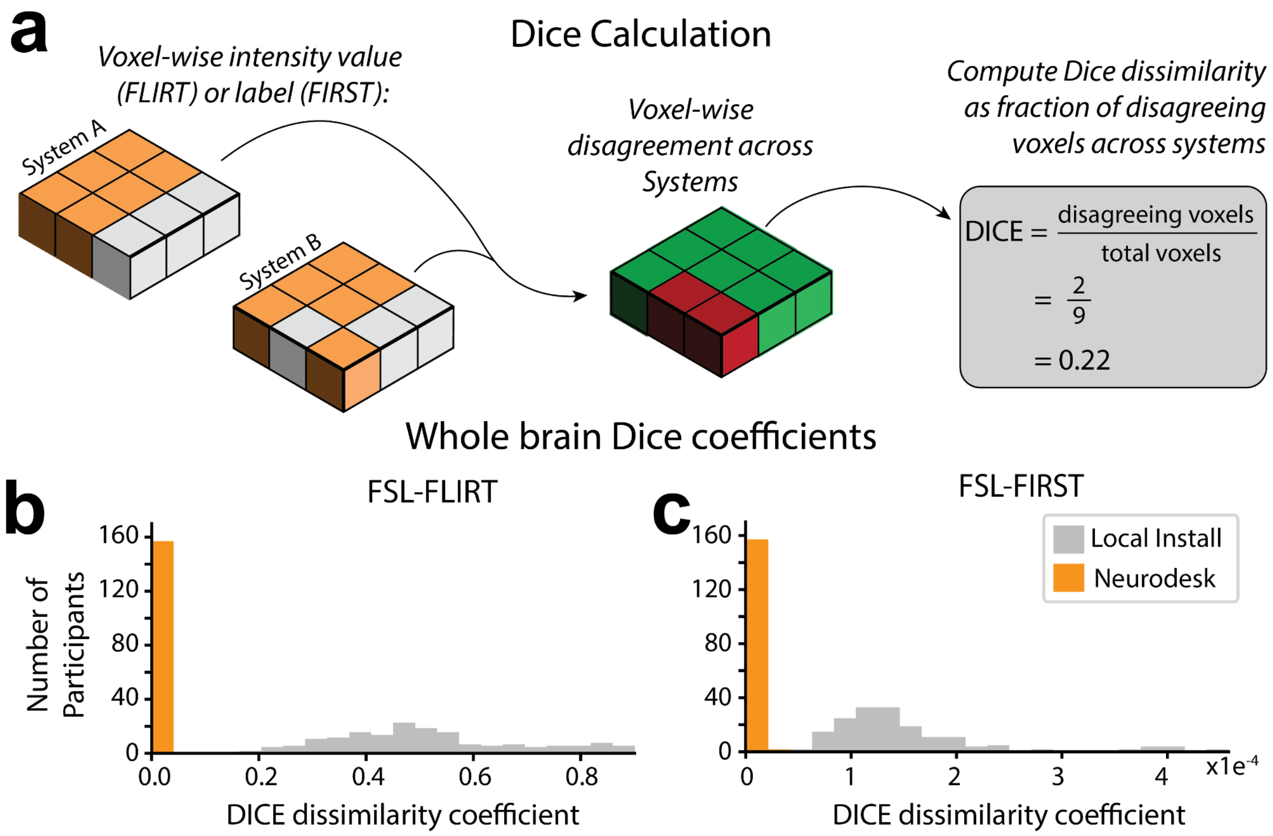
Scientific progress fundamentally depends on the peer review process - scientists must be able to critically assess reported findings and conclusions based on a clear and thorough methodological description[1]. Well-documented experimental code is the most thorough description of any analysis pipeline. However, differences in computing environments and dependencies mean that access to this source code does not guarantee the capability to run the code nor arrive at the same result[2,3]. Reproducibility has therefore come to represent a minimum standard by which to judge scientific claims[1,2,4]. Unfortunately, scientific reproducibility is often not attainable due to differences in the outcomes of neuroimaging pipelines across different computing environments, as previously documented[5–7]. Glatard et al. (2015) demonstrated this effect for several MRI analysis pipelines, showing that differences in the implementation of floating-point arithmetic across operating systems accumulated throughout long analysis pipelines and led to meaningful differences in the results[5]. Neurodesk solves this issue using 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[8] was installed both locally and within Neurodesk on two separate computers (System A, System B) which were running different GNU/Linux distributions. This resulted in four unique computing environments (see **Supplementary Table 1**). Glatard et al.'s FSL-based analyses, namely the brain extraction (Brain Extraction Tool [FSL-BET]), 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)[9]. 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 intra- and inter-environment comparisons, we first compared file checksums. When two files produced different checksums, we quantified the pairwise differences across systems by computing Dice dissimilarity coefficients across images (**Supplementary Figure 1a**). Note that there were never any intra-system differences in checksums (i.e., all analyses were deterministic, resulting in identical outcomes when run twice in the same computing environment). The code used to implement these analyses is available and re-executable through Neurodesk Play at: https://osf.io/e6pw3/.

Supplementary Table 1. 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	



Supplementary Figure 1. Discrepancies in image registration and tissue segmentation. (**a**) Calculation of the Dice dissimilarity coefficients: for each image, the voxel-wise image intensity (FLIRT) or label (FIRST) were calculated on System A vs System B, then for each participant the fraction of disagreeing voxels out of the total number of voxels was calculated. Disagreement meant the same voxels had a different intensity after image registration on System A vs. System B. Thus a Dice coefficient of 0 indicates the outputs were identical. (**b**) Histograms of Dice dissimilarity coefficients for image intensity calculated with FSL-FLIRT on Neurodesk vs. Local Install. A Dice coefficient of 0 indicates that the image intensity was matched across both systems at every voxel, while Dice coefficients of 1 indicate that no voxels matched in value across the two systems. (**c**) Histograms of Dice dissimilarity coefficients for subcortical structure labels calculated using FSL-FIRST on Neurodesk vs. Local install. Here, “disagreement”, for the purpose of DICE calculation, 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 much smaller than for image registration. This is expected because there are 73 times more “classes” for the image registration task, which uses image intensity (Range: 0 − 1903) as a class, than the classification task, which has labels for 15 structures. *Note*: The first bin in each histogram spans values greater than 0. See text for exact values of DICE coefficients.

Brain extraction and tissue classification

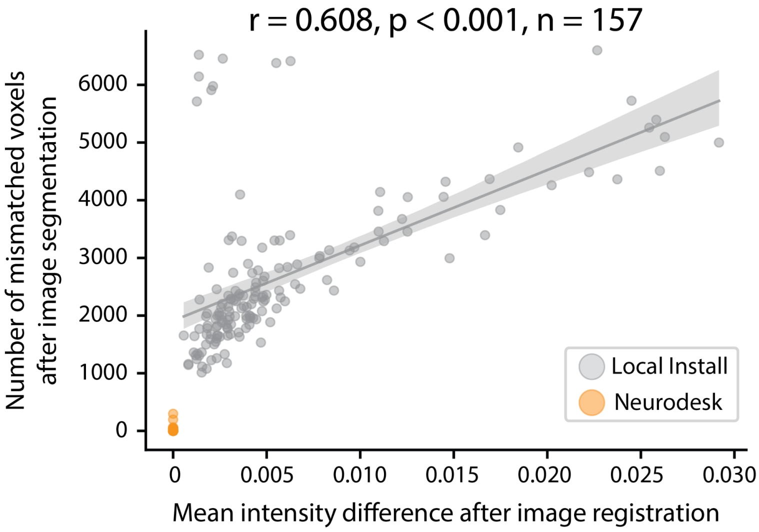
FSL BET and FAST were run on raw MRI images to extract voxels containing brain tissue and classify tissue types, respectively. The file checksums for the outputs of these processing steps were identical across all computing environments, verifying that the implementation of the processing pipeline was reproducible across systems for both Neurodesk and local installation up to this point.

Image registration

FSL FLIRT was applied to register the images to the standard MNI-152 T1 1 mm template using 12 degrees of freedom. Dice dissimilarity coefficients for each image were computed to quantify the pairwise differences in image intensity across systems (**Supplementary Figure 1a**). 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, **Supplementary Figure 1b**). These high Dice dissimilarity coefficients for the local installation indicate differences across many voxels. However, the magnitude of these differences in image intensity was subtle (inter-system intensity difference; M = 1.88, SD = 1.97; where $intensity \in Z: intensity \in [0, 1903]$, **Figure 2a, b**).

Subcortical tissue segmentation

Differences in image intensity across local installations were widespread yet 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 for 93% of images when run with Neurodesk. Computation of the Dice dissimilarity coefficients for each type of installation revealed that while differences were small, they had non-overlapping ranges. Indeed, differences were much less prevalent for the Neurodesk installations (Dice dissimilarity coefficient; Range: 0.00 – 2.20x10^{-5}, M = 3.43x10^{-7}, SD < 0.01) compared with the local installations (Dice dissimilarity coefficient; Range: 5.80x10^{-5} – 4.59x10^{-4}, M = 1.46x10^{-4}, SD < 0.01, **Supplementary Figure 1c**). On average, there were 426 times more voxel-wise disagreements 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 2c, d**). These projections illustrate that differences for locally installed software were widespread across all subcortical structures (**Figure 2c**), while any subtle differences for Neurodesk were limited to a few voxels (**Figure 2d**).

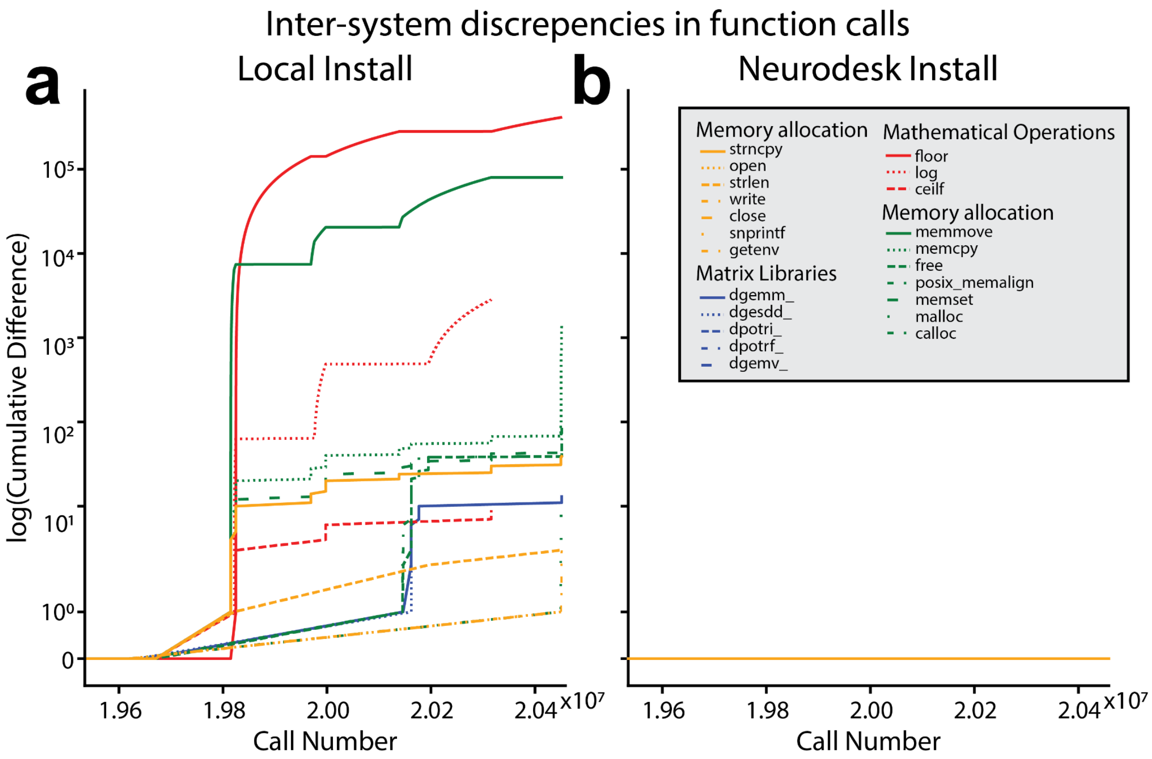


Supplementary Figure 2. Scatter plot showing the mean inter-system image intensity differences across all voxels within the classified subcortical structures vs. the number of voxels subsequently classified with different labels across systems, a positive Pearson's correlation was observed (r = 0.608, n = 157, two-sided p = 2.905e-17, 95% CI [0.499, 0.698]). Shaded area represents the 95% confidence interval around the line of best fit for the local installation data.

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 $r = 0.608$, $p < .001$, **Supplementary Figure 2**). 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 (**Supplementary Figure 3**). This leads to 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[10,11]. Reasons include whether 64 (SIMD, GPU) or 80 bit (x87 FPU) precision is used internally, reduced rounding for fused multiply-add, or if negative zero and positive zero are considered equal. Critically, these differences are minor, which is likely why the differences in classification across systems for Neurodesk were subtle.



Supplementary Figure S3. 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 occurred earlier in time and the discrepancies for *floorf()* far outnumbered those for any other function from the locally installed tool.

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 links 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 addressed 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 (**Supplementary Table 2**). The *floorf()* function represented the most prevalent difference in library calls. There were over 13 000 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. 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 (**Supplementary Figure 3a**). However, all remaining mismatches across systems occurred in memory allocation functions. Notably, the Neurodesk implementation of FSL (**Supplementary Figure 3b**) ran across systems with no differences in floating point representation or calls to shared libraries, while maintaining a similar runtime to local installation on the same hardware (**Supplementary Table 2**).

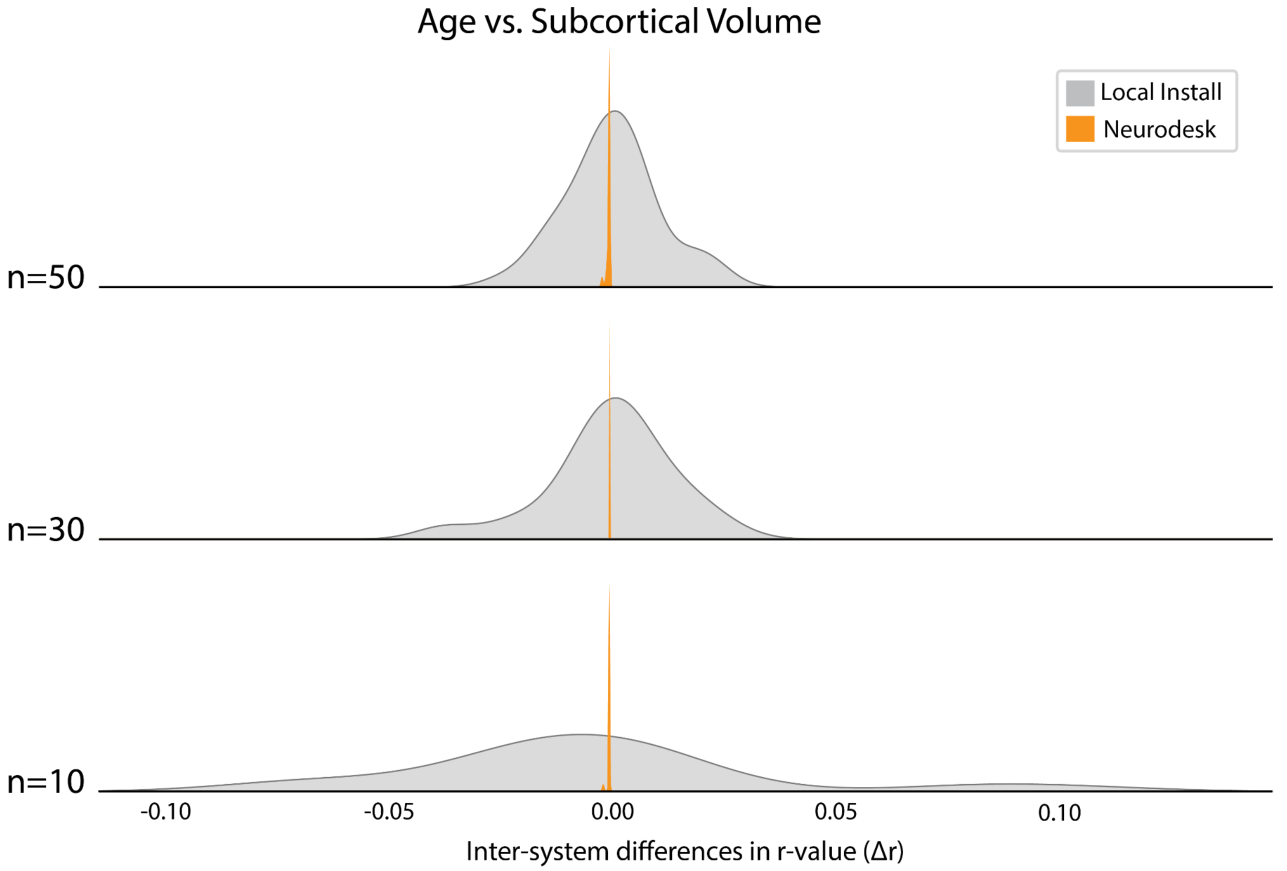
Supplementary Table 2. Differences in the execution of tissue segmentation (FIRST) and image registration (FLIRT) pipelines. Runtime refers to the CPU time spent on system and library calls within a pipeline.

	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=8)	**System A**	**System B**	**System A**	**System B**
Average (mins)	4.88	5.39	5.73	5.47
Standard Deviation (mins)	0.07	0.19	0.20	0.15

Understanding the practical implications of inter-system differences.

The local installations led to inter-system differences in tissue classification 3 orders of magnitude larger than in Neurodesk. However, it is difficult to know how voxel-wise differences of this scale might actually affect test statistics, i.e. could there actually be a different conclusion about the research question if the same analysis on the same data runs on a different computer[12]? 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 permutation tests for each of these volumes (9999 permutations each). Each permutation consisted of randomly selecting a fixed number of samples (three different sample sizes were used, n=10, 30, 50), followed by computing the Pearson correlation of volume vs. participant age. Finally, we calculated the differences in the correlation scores across the two systems. Critically, for each permutation, the same sample was used across 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 209979 permuted samples (7 subcortical structures (Putamen, Amygdala, Thalamus, Pallidum, Caudate Nucleus, Hippocampus, Accumbens.) x 3 regions (left hemisphere, right hemisphere, both) x 9999 subject-wise permutations).



Supplementary Figure 4. 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).

The analysis showed that as sample size decreased, the inter-system correlation score 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; **Supplementary Figure 4**). By contrast, the inter-system correlation score differences for Neurodesk were negligible and did not scale with sample size (Neurodesk: N=50, Δr = $-1.74\text{x}10^{-3}$ – $2.59\text{x}10^{-4}$ | N=30, Δr = $-3.75\text{x}10^{-5}$ – $1.89\text{x}10^{-4}$ | N=10, Δr = $-1.52\text{x}10^{-3}$ – 0; **Supplementary Figure 4**). 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 variability when conducting these types of analyses.

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

Frequently Asked Questions

How could researchers build an analysis pipeline and share this with other researchers using Neurodesk?

We provide computational notebook examples to showcase how different tools can be used in fully reproducible and shareable analysis pipelines: https://www.neurodesk.org/tutorials-examples/examples/. In these examples, we demonstrate the use of various tools on publicly available datasets. We used the open-source nipype workflow system to execute analyses on this data, enabling complex analyses to be built, shared, and executed identically across Neurodesk installations.

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 your analysis speed the first time you use a new tool. Neurodesk uses the CernVM File System (CVMFS), meaning that only the specific part of a container that 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 **Supplementary table 2**). Although there is a container initialisation time that could impact performance in comparison to a non-containerised 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[13].

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. We aim to get mirror servers as close as possible to all users so that CVMFS can automatically use the fastest available mirror server.

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 the least privileges to minimise the impact of malicious software. Interacting with the web from within a Neurodesktop poses a similar risk to any system with access to the internet, so all precautions would apply. Neurodesktop can be shut down, deleted and started fresh with minimal effort, which means recovery is substantially simpler than a native installation in a similar scenario. To ensure data security, it is essential for users who run Neurodesk on a cloud provider or in their local network to follow security best practices and secure the port Neurodesktop is running on via firewall rules. For an in-depth review of the potential security concerns of containerising scientific data analysis software, see Kaur et al. (2021)[14].

Can I store processed data in Neurodesk?

Neurodesktop allows host directories to be mounted for local data access, and these directories can then be accessed from the Neurocontainers. Data can also be accessed via access clients and the web inside a Neurodesktop instance running for example on a cloud provider. Upon installation of Neurodesk on a local PC or HPC, users have the option of mounting an existing local directory or utilising the automatically created and locally stored directory, ~/neurodesktop-storage. This directory is permanently stored on the local host and will remain even if Neurodesktop is deleted, ensuring that the data remains on the local host and does not leave their computer. It is important to note that the data remains on the user's computer if Neurodesk is running locally, but Neurodesk can also run in a cloud environment where data is stored remotely and users need to ensure that their use case is in line with their ethics and data agreements.

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

Neurodesktop is a Docker container packaging a GNU/Linux desktop environment that delivers neuroscience applications via CVMFS, distributed through 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, including 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 (https://lmod.readthedocs.io/en/latest/); 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 provided 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 the 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 extending Neurodesk's current functionality could be covered by charging a nominal fee to manage the resources required to deploy Neurodesk in combination with Jupyterhub 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 GNU/Linux distribution, 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.

References

1. Plesser, H. E. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Front. Neuroinformatics* **11**, 76 (2018).
2. Peng, R. D. Reproducible Research in Computational Science. *Science* **334**, 1226–1227 (2011).
3. 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).
4. 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).
5. Glatard, T. *et al.* Reproducibility of neuroimaging analyses across operating systems. *Front. Neuroinformatics* **9**, 12 (2015).
6. 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).
7. 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).
8. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M. FSL.

NeuroImage **62**, 782–790 (2012).

9. 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).
10. The pitfalls of verifying floating-point computations | ACM Transactions on Programming Languages and Systems. https://dl.acm.org/doi/abs/10.1145/1353445.1353446.
11. Gorgolewski, K. J. *et al.* BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. *PLOS Comput. Biol.* **13**, e1005209 (2017).
12. Kiar, G. *et al.* Numerical Uncertainty in Analytical Pipelines Lead to Impactful Variability in Brain Networks. 2020.10.15.341495 Preprint at https://doi.org/10.1101/2020.10.15.341495 (2021).
13. Rioux, P., Kiar, G., Hutton, A., Evans, A. C. & Brown, S. T. Deploying large fixed file datasets with SquashFS and Singularity. in *Practice and Experience in Advanced Research Computing* 72–76 (Association for Computing Machinery, 2020). doi:10.1145/3311790.3401776.
14. Kaur, B., Dugré, M., Hanna, A. & Glatard, T. An analysis of security vulnerabilities in container images for scientific data analysis. *GigaScience* **10**, giab025 (2021).