Chapter 8

Standardized Preprocessing in Neuroimaging: Enhancing Reliability and Reproducibility

Oscar Esteban

Abstract

This chapter critically examines the standardization of preprocessing in neuroimaging, exploring the field's evolution, the necessity of methodological consistency, and the future directions shaped by artificial intelligence (AI). It begins with an overview of the technical advancements and the emergence of software tools with standardized neuroimaging processes. It also emphasizes the importance of the *Brain Imaging Data Structure* (*BIDS*) and data sharing to improve reproducibility. The chapter then discusses the impact of methodological choices on research reliability, advocating for standardization to mitigate analytical variability.

The multifaceted approach to standardization is explored, including workflow architecture, quality control, and community involvement in open-source projects. Challenges such as method selection, resource optimization, and the integration of AI are addressed, highlighting the role of openly available data and the potential of AI-assisted code writing in enhancing productivity.

In conclusion, the chapter underscores *NiPreps*' contribution to providing reliable and reproducible preprocessing solutions, inviting community engagement to advance neuroimaging research. The chapter envisions a collaborative and robust scientific culture in neuroimaging by promoting standardized practices.

Key words Computational neuroscience, fMRIPrep, MRIQC, Neuroimaging, Reliability, Reproducibility, Python, Open-source

1 Introduction

Neuroimaging has seen remarkable technical developments over the past three decades, reflecting its singular adequacy for probing the brain's structure and its intricate workings in vivo. The continuous innovation in image formation technologies has bolstered the development of domain software tools and applications. Correspondingly, new theories of the brain and new experimental approaches have also stimulated progress with the demand for new hardware and software instruments [1]. Consequently, the field has produced a multiplicity of software instruments over time, developed with high engineering standards and readily

Robert Whelan and Hervé Lemaître (eds.), *Methods for Analyzing Large Neuroimaging Datasets*, Neuromethods, vol. 218, https://doi.org/10.1007/978-1-0716-4260-3_8,

available to neuroimagers. Among many others, *AFNI* [2], *FreeSurfer* [3], *FSL* [4], and *SPM* [5] have achieved remarkable adoption. For transparency, most neuroimaging packages enable researchers to independently scrutinize the implementations they add to their tool belts by making the source code accessible—if not fully open-source.[1]

With the advancement of imaging techniques, these toolboxes have been substantially expanded to support ever-growing spatial, and temporal resolutions, as well as new modalities and acquisition approaches. Moreover, neuroimaging research has also seeded convergent efforts toward multimodal fusion, where features are extracted from several measurement types. These two drivers prompted the establishment of neuroimaging "pipelines" that stage processing steps and encompass data management operations. Traditionally, these steps are drawn from a single toolbox of choice for a particular application or analysis, as compatibility between tools poses substantial problems. Tools such as *Nibabel* [6] and *Nipype* [7] have enabled "mixing-and-matching" from available neuroimaging tools to select the best-in-class implementations across them by standardizing access to data (*Nibabel*) and the user interface to tools (*Nipype*).

While the redundancy of implementations for a given task is positive from a knowledge formalization and accessibility perspective, it has gradually become apparent that methodological variability is an obstacle to obtaining reliable results and interpretations, a problem only exacerbated by inaccurate or insufficient reporting [8]. Indeed, variations in processing methods across different modalities, research groups, studies, and even individual researchers have contributed to inconsistencies and discrepancies in reported findings [9–11]. This problem was more recently surfaced with the *Neuroimaging Analysis Replication and Prediction Study* (*NARPS*; [12]), where 70 teams of functional MRI (fMRI) experts were provided with the same dataset and tasked with testing a closed set of nine hypotheses. The results highlighted an overall poor agreement in conclusions across teams. Considering that no two teams fully coincided in the design of their analysis pipelines, the *NARPS* authors interpreted that methodological variability was at the core of the divergent results. As potential counter-measures, Botvinik-Nezer and colleagues discussed the value of preregistration to avoid methodological variability introduced *post-hoc*, that is, fine-tuning the processing pipeline until the results align with expectations. Additionally, they also envisioned multiverse analyses where many combinations of different implementations of a pro-

[1] A source code may be made accessible (e.g., shared over a private email) while open-source implies a license stating unambiguous terms for reuse and redistribution.

cessing and analysis pipeline are explored, and results are either interpreted as a range of possibilities or aggregated statistically [13], e.g., by means of active learning [14]. Both—preregistration and multiverse analyses—are powerful tools for reproducibility that operate in the domain of methodologies, either limiting the researcher's degrees of freedom and incentives to workaround nonnegative findings (preregistration; [15, 16]) or embracing the exploration of the breadth of methodological alternatives and combinations thereof (multiverse; [17]). Because shallow reporting bears great responsibility for how analytical variability may undermine the reliability, best practices in reporting such as checklists (for instance, the Organization for the Human Brain Mapping's *COBIDAS*; *Committee on Best Practice in Data Analysis and Sharing*; [18]) have been proposed to solve the problem. Nonetheless, it is worth noting that all the teams involved in *NARPS* completed the *COBIDAS* checklist. That alone did not guarantee that the reported methods could be adequately replicated. Taylor and colleagues showed evidence that the *NARPS* results are more convergent than initially interpreted when outputs are examined without standardly applied simplifications such as thresholding of statistical maps [19]. Nonetheless, analytical variability remains a concerning issue that undermines the reliability of neuroimaging research.

Over the last decade, researchers have harnessed their neuroimaging workflows targeting reliability. The *Brain Imaging Data Structure* (*BIDS*; [20, 21]) has proven to be a hallmark example of how standard dataset organization is critical to implement reproducible neuroimaging research (*see* Chapter 4 for a detailed guide to *BIDS*). Not only has *BIDS* deeply transformed the neuroimaging landscape by establishing a consistent agreement on how data and metadata must be organized, maximizing the shareability of datasets and ensuring proper data archiving, it has also spurred a body of research addressing aforementioned challenges to reproducibility. Indeed, data sharing has been recognized as a powerful reproducibility tool, and outstanding resources such as *OpenNeuro* [22] have contributed to solidifying the development of neuroimaging workflows with a clear and standardized interface for input—*BIDS*— and output—*BIDS-Derivatives*—data. Leveraging *BIDS* and following the *BIDS Apps* principles [23], our *fMRIPrep* application [24] has shaped the development of standardized neuroimaging workflows and given rise to the *NeuroImaging PREProcessing toolS* (*NiPreps*; [25]). Using the *NiPreps* development experience as a foundation, this chapter explores the rationale, benefits, and potential trade-offs of standardizing the preprocessing stage as a way to account for analytical variability in a significant stage of every neuroimaging pipeline.

2 Standardizing Preprocessing: What and Why?

Generally, neuroimaging analyses cannot be carried out directly with "unprocessed" data, that is data after reconstruction from the "raw" recordings collected by an imaging device (c.f., Chapter 15). While *BIDS* helps organize unprocessed data and provides a reliable ingress interface into subsequent processing, data needs preprocessing before it can be analyzed [25, 26]. Preprocessing involves a series of essential operations, including data cleaning, spatiotemporal normalization and alignment, artifact removal, and other steps required by statistical modeling [24]. Analytical variability quickly emerges in the design of such pipelines as each processing step with its associated parameters involves methodological choices. These choices will likely undermine the reliability of the outcomes unless the pipeline abides by strict self-consistency and transparent implementation and reporting. The variability introduced by preprocessing compounds with the variability of the data collection and subsequent steps, such as image reconstruction, exacerbating the overall pipeline's unreliability.

When looking through the lens of classical test theory [27–29], the neuroimaging "scores" that are statistically modeled in the final analysis step are indeed "preprocessed" data. For example, a morphometry analysis quantifying T_1-weighted MRI properties such as cortical thickness—that is, a neuroimaging "score",— requires preprocessing steps involving brain extraction or reconstruction of brain surfaces. The classical theory posits two approaches to improve the reliability of scores, such as the cortical thickness in the example: aggregation and standardization. Please note that although "reliability", "reproducibility", "repeatability", and "replication" are different terms understood in many ways across disciplines [30], here we will define reliability as the property that the score or measurement is consistently correlated with the true value of it [29].

The aggregation approach follows the "Spearman-Brown prophecy formula" [31, 32], and supports that random error components cancel out by aggregating items of the same true score, thereby providing more reliable measurements. This aggregation approach is at the core of recent dense and "personalized" imaging data collection efforts [33]. These approaches repeat the same experiment on reduced numbers of individuals to analyze them independently and focus only on within-subject variability, thereby improving the within-individual reliability. Within the example of cortical thickness analysis, aggregation could be implemented by collecting several images for every subject and extracting surfaces and the feature of interest from each individual's single, average template. The approach is an excellent tool to characterize the reliability of the measurements (*see* ref. [34], for an example of

our efforts in this direction). On the other hand, standardization reduces sources of variability relating to the measurement instrumentation, including methodological variability of preprocessing, by strictly predetermining all the experimental choices and establishing a unique workflow. This chapter focuses on standardization to reduce the domain of coexisting analytical alternatives, hence reducing the multiverse that must be traversed in mapping the variability of the results. Standardizing the preprocessing offers numerous benefits for enhancing the reliability and reproducibility of the research workflow, albeit the paradigm is not free of trade-offs and challenges. First, a reliable measure is not necessarily "valid" [35]. Standardization may enforce specific assumptions about the data and introduce biases that could void the measurement validity. For instance, a brain extraction algorithm within a brain morphometry application that systematically includes dura in smaller brains and excludes it in larger ones could lead to the wrong conclusion about gender differences in cortical thickness at the population level. Other challenges and trade-offs of standardization involve the robustness to data diversity, the flexibility versus experimental degrees-of-freedom trade-off, and computational optimization. The following section explores several of these aspects along different dimensions through which standardization may be implemented.

3 Dimensions of Standardization

3.1 The Brain Imaging Data Structure (BIDS)

Although initially conceived as an exchange format to maximize data shareability and archival, *BIDS* provides a consistent framework for structuring data directories, naming conventions, and metadata specifications. Building on the clear interface that *BIDS* affords for the input, our *BIDS Apps* framework [23] describes several formal aspects to enable the standardization of pipelines. The widespread adoption of *BIDS* has greatly facilitated the uptake of *BIDS Apps*, such as *fMRIPrep*, which leverages *BIDS*-compliant datasets to automate the preprocessing of fMRI and exchange (through *BIDS-Derivatives*) downstream processing and analysis. *BIDS-Derivatives* provides a standardized format for representing processed and derived data, ensuring consistency and compatibility across different studies and analyses. Researchers can easily share and disseminate their preprocessed data by employing *BIDS-Derivatives*, enabling reproducibility and promoting collaboration within the neuroimaging community. The *BIDS* specification has permitted the development of tooling such as the *PyBIDS* library [36] to query and retrieve data and metadata from the input dataset and generate the names and structure of the final derivatives at the output (*see* Note and Chapter 2).

Note

***PyBIDS* is a *Python* package that makes it easier to work with *BIDS* datasets.** At present, its core and most widely used module supports simple and flexible querying and manipulation of *BIDS* datasets. *PyBIDS* makes it easy for researchers and developers working in *Python* to search for *BIDS* files by keywords and/or metadata; to consolidate and retrieve file-associated metadata spread out across multiple levels of a *BIDS* hierarchy; to construct *BIDS*-valid path names for new files; and to validate projects against the *BIDS* specification, among other applications. For further details on its core indexing module and other additional utilities it provides, *see* ref. [36].

First, we show how to pre-index the dataset. This will speed up later querying and is especially time-saving when the dataset is sizeable. Once *PyBIDS* is installed (*see* its documentation website for instructions, https://bids-standard.github.io/pybids/; *see* Resources), issue the following command from within the directory in which data are available (e.g., installed with *DataLad*):

```
cd /data/ds002790
mkdir -p .bids-index/
pybids layout --reset-db --no-validate --index-metadata .
.bids-index/
```

Once the dataset is indexed, we can open a *Jupyter notebook* or an *IPython* console and explore the dataset. We first import *PyBIDS* (with the package name "`bids`" within the *Python* distribution) and create a *dataset layout* object called `ds002790`. We make sure to point it to the right folder and to use the index database created above:

```
>>> import bids
>>> ds002790 = bids.BIDSLayout(
... "/data/ds002790",
... database_path="/data/ds002790/.bids-index/",
... )
```

We now use the *dataset layout* to query the dataset. In general, *PyBIDS* enables querying for metadata using `get_<metadata-name>` calls. For instance, we can check the total number of subjects:

```
>>> len(ds002790.get_subjects())
226
```

(continued)

We can also investigate the data and metadata types by querying the available BIDS' suffixes:

```
>>> ds002790.get_suffixes()
["T1w", "description", "dwi", "participants", "magni-
tude1", "phasediff", "bold", "physio", "events"]
```

We can also query for metadata entries, and filtering by a given suffix ("`bold`") discover that all BOLD images have a repetition time of 2.0 s:

```
>>> ds002790.get_RepetitionTime(suffix="bold")
[2]
```

Now we list the four BOLD fMRI tasks in the dataset:

```
>>> ds002790.get_tasks(suffix="bold")
["restingstate", "stopsignal", "workingmemory", "emo-
matching"]
```

or get the path to all *NIfTI* files corresponding to the resting-state BOLD runs for only the first subject (here only one file):

```
>>> ds002790.get(
... subject=ds002790.get_subjects()[0],
... task="restingstate",
... suffix="bold",
... extension=[".nii", ".nii.gz"],
... )
[<BIDSImageFile filename='/data/datasets/ds002790/sub-
0001/func/sub-0001_task-restingstate_acq-seq_bold.nii.
gz'>]
```

Notably, the *BIDS* and *BIDS-Derivatives* specifications allow *BIDS Apps*, such as *MRIQC* [37] and *fMRIPrep*, to follow a simple pattern for their invocation from the command line (*see* Note).

Note
Standardized command line of *BIDS Apps*. A *BIDS App* is a container image capturing a neuroimaging pipeline that takes a *BIDS*-formatted dataset as input. Since the input is a whole dataset, apps are able to combine multiple modalities, sessions, and/or subjects, but at the same time, need to implement ways to query input datasets. Each *BIDS App* has the

(continued)

same core set of command-line arguments, making them easy to run and integrate into automated platforms. *BIDS Apps* are constructed in a way that does not depend on any software outside of the container image other than the container engine. Further documentation about *BIDS Apps* and their execution with containers is found at the *NiPreps* website (https://www.nipreps.org/apps/framework/; *see* also Resources). An index of *BIDS Apps* is maintained at https://bids-apps.neuroimaging.io/apps/ (*see* Resources).

All *BIDS Apps* share a common command line interface that enables their automated concatenation and execution. The command line follows the structure `runscript input_dataset output_folder analysis_level <optional named arguments>`, where `runscript` is typically the name of the *BIDS App* (e.g., `mriqc`), `input_dataset` points at the path of the input *BIDS* or *BIDS-Derivatives* dataset, `output_folder` points at the path where results will be stored, and `analysis_level` can be either `participant` or `group` depending on what type of analysis will be executed, as introduced in Fig. 1. Therefore, the general structure of the command line is particularized for executing *MRIQC* as follows:

```
mriqc /data/ds002790 /data/ds002790/derivatives/
mriqc_24.0.0 participant
```

Following the *BIDS-Derivatives* specifications developed after the *BIDS Apps* framework, the output directory is set to `/data/ds002790/derivatives/mriqc_24.0.0`, which can naturally be managed as a *DataLad* "subdataset" and indicates that the choice of *MRIQC*'s release is 24.0.0 (*see* Subheading 3.4 for standardization of versioning). We will leverage *PyBIDS*' index cache by adding one named argument to the baseline command line:

```
mriqc /data/ds002790 /data/ds002790/derivatives/
mriqc_24.0.0 participant \
 --bids-database-dir /data/ds002790/.bids-index/
```

where `--bids-database-dir /data/ds002790/.bids-index/` is an optional argument *MRIQC* accepts to employ a pre-indexed database. The command line naturally generalizes to *fMRIPrep* as follows:

```
fmriprep /data/ds002790 /data/ds002790/derivatives/fmri-
prep_24.0.0 participant \
 --bids-database-dir /data/ds002790/.bids-index/
```

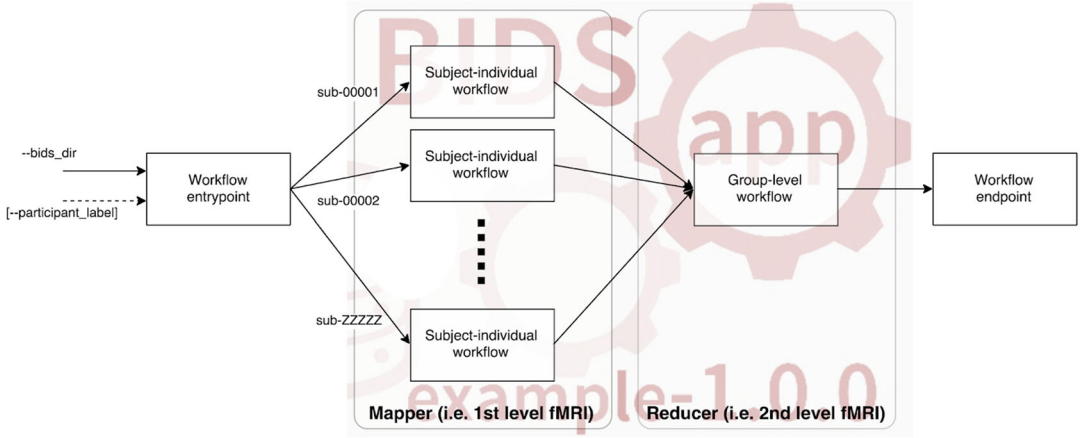

Fig. 1 *BIDS Apps* made strides toward standardization of design. In particular, our manuscript elaborated that most neuroimaging applications could be modularized in a first step ("mapper"), where independent processes are executed, followed by a second step ("reducer"), where results from the previous stage are aggregated

3.2 Standardization of Design

Workflow architecture *BIDS Apps* also promoted the standardization of workflow design. Gorgolewski and colleagues [23] described two typical execution patterns in neuroimaging analyses (Fig. 1). First, some workflows like *fMRIPrep* focus on individual subjects, where subjects' processing can be "embarrassingly parallel" thanks to the independence between execution processes. Generally, data from individual subjects (*participant level*) are then aggregated and compared depending on the study design (*group level*). In group-level analyses, inter-dependencies in the compute graph disallow embarrassingly parallel approaches. Therefore, parallelization must be implemented either at the level of task, either within a computing node (e.g., threading, multiprocessing, GPU, etc.) or across computing nodes (e.g., with message passing interface). By focusing only on the participant level, *fMRIPrep*, *MRIQC*, or any other *NiPreps* can optimize the workflow for the specific execution mode.

Building from the *BIDS Apps* standard, *fMRIPrep*, and *MRIQC* continued developing standardizations of design that would evolve into the *NiPreps* framework. Beyond sharing the same infrastructure to handle *BIDS*, the modularity of workflows, or the use of *NiPype* as the workflow engine, an alpha release of *fMRIPrep* now adopts a "fit and transform" paradigm, inspired by *Scikit-Learn*'s influential interface in machine learning [38]. Under this paradigm, *fMRIPrep* generates only a minimal set of results from which the "traditional" outputs of *fMRIPrep* can deterministically be generated. The minimal set of results includes linear and nonlinear spatial mappings between coordinate systems of interest (anatomical

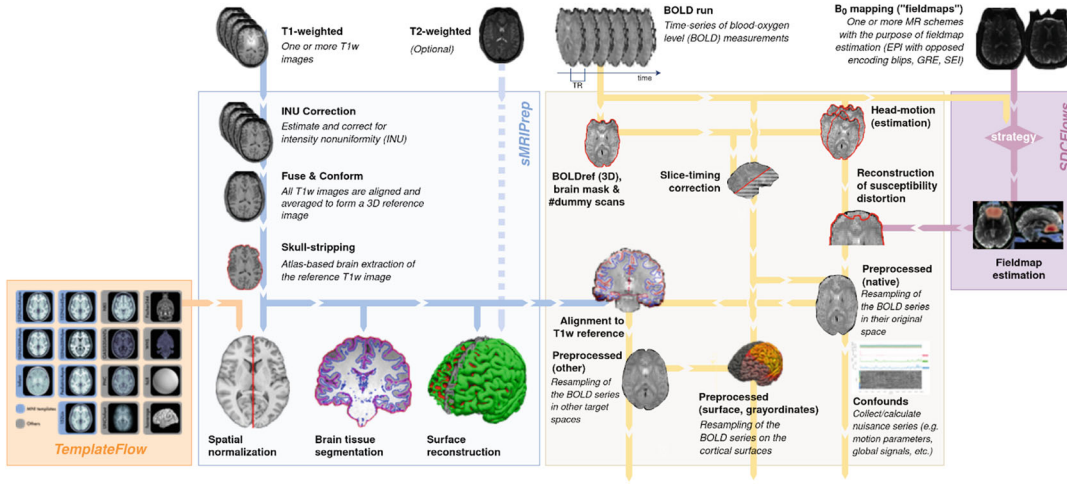

Fig. 2 *fMRIPrep* underwent a "deconstruction" effort, giving rise to *NiPreps*. We identified several preprocessing steps that required further standardization beyond fMRI. In particular, modularizing *fMRIPrep* derived in two relevant *NiPreps*—*TemplateFlow* [41], for the standardization of templates and atlases; and *SDCFlows* [42], which contains workflows and tools for the estimation and correction of susceptibility-derived distortions of echo-planar images (EPI) that are commonly employed to acquire fMRI and dMRI data

image, functional images, standard space defined by templates, etc.), temporal mappings (e.g., slice-time information), and estimation of spatiotemporal artifacts (e.g., susceptibility distortion, head-motion, etc.), associated with the processing steps of Fig. 2. This division of the workload benefits researchers and data stewards to maximize the value of the shared data as downstream users can generate the desired *fMRIPrep* results in the spatial frame they need while minimizing data transfer and storage. Indeed, we evaluated that the new approach resulted in 25–52%, 43–54%, and 72–87% reductions in runtime, data volume, and file counts, respectively, in comparison to the previous version of *fMRIPrep* [39]. The paradigm also facilitates the generalizability of software implementation, as only the fit step requires adaptation across modalities (e.g., *dMRIPrep* to preprocess diffusion MRI, dMRI), populations (e.g., *fMRIPrep-infants*; [40]), and species (e.g., *fMRIPrep-rodents*) while the transform step can have a single approach thereby reducing maintenance and technical debt.

Building blocks and modularity Based on the success story of *fMRIPrep*, we initiated a focus on the generalization of the workflow, first within blood-oxygen-level-dependent (BOLD) fMRI for its application on infants (*fMRIPrep-infants*) and rodents (*MRIQC-rodents*; [43]; and *fMRIPrep-rodents*). Similarly, the development of "-Preps" for other modalities was initiated. Indeed, *dMRIPrep* is the counterpart of *fMRIPrep* for dMRI data [44], *ASLPrep* covers arterial-spin-labeling (ASL; [45]) fMRI, and

PETPrep targets positron emission tomography (PET) imaging. These generalizations were soon denominated as *NeuroImaging PREProcessing toolS* (*NiPreps*), with the overarching goal of standardizing preprocessing components shared across modalities, populations, and even species. *NiPreps* adopts a modular and extensible architecture, allowing researchers to combine and configure different preprocessing modules to suit their specific needs (Figure 3). Modularity enables *fMRIPrep* and *dMRIPrep* to share the preprocessing of structural imaging (T_1-weighted and T_2-weighted) through *sMRIPrep*, and susceptibility distortion mapping through *SDCFlows* [42]. Therefore, the results of multimodal (f/dMRI) studies are referred to a single *version* of the anatomy obtained with a single run of *sMRIPrep* and distortions

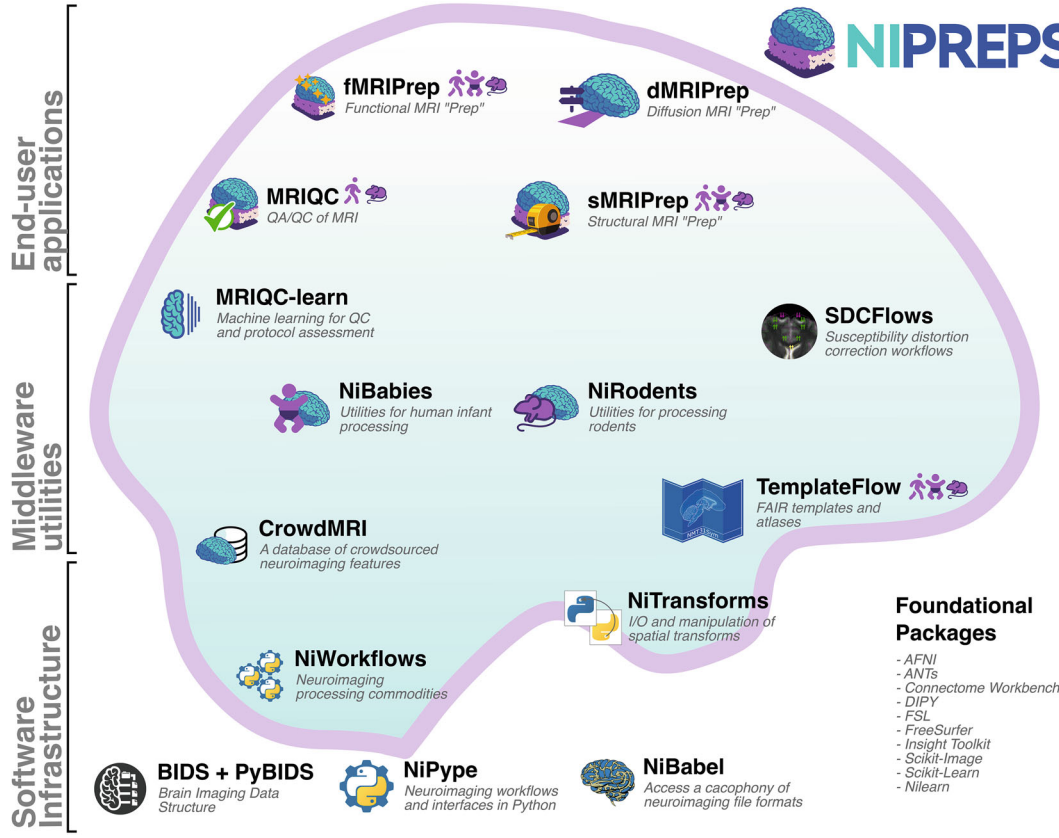

Fig. 3 The *NeuroImaging PREProcessing toolS* (*NiPreps*) framework. The *NiPreps* framework encompasses modular neuroimaging software projects. A number of projects provide the infrastructure over which more elaborate or abstract features are implemented. Leveraging that base, end-user applications such as *fMRIPrep*, *dMRIPrep*, or *MRIQC* can be developed. Projects outside the brain edge (*Nipype*, *Nibabel*, and *BIDS*/*PyBIDS*) are not part of the framework but receive upstream contributions and are essential software foundations for the whole vision. Finally, foundational packages support the most basic algorithmic implementation of methods

of the echo-planar imaging (typically employed to acquire diffusion and functional MRI) are addressed consistently. While this example showcases modularization at the highest level of the software stack shown in Fig. 3, the principle applies to the smaller components at lower levels of abstraction.

Quality assessment and control (QA/QC) Deploying a QA/QC strategy is critical for the reproducibility of the neuroimaging workflow [26]. In addition to increasing the workflow's and results' reliability, standardizing QA/QC is critical to ensure quality issues do not propagate unidentified along studies. Robust QA/QC implies setting up several QA/QC checkpoints along the neuroimaging pipeline to ensure that data meeting exclusion criteria are dismissed before reaching analysis [46]. Under such a definition, QC checkpoints are analogous to the layers of the so-called "Swiss cheese security model" [47], with the goal that data of insufficient quality which may bias the results does not reach the analysis. *NiPreps* such as *MRIQC* and *fMRIPrep* generate visual reports to implement standardized QA/QC protocols. In *MRIQC*, visual reports enable one mechanism for assessing the quality of outcomes of an experimental session, that is, the "original" unprocessed data. In the case of downstream pipelines such as *s/d/fMRIPrep*, the objective of the checkpoint is to ensure the preprocessing is fit to the study requirements. We standardized report generation in *NiPreps* by outsourcing *NiReports* (*NeuroImaging Reports*) as a standalone library independent of *fMRIPrep*'s codebase. *NiReports* comprises two basic components: unitary visual elements or "reportlets" and the "assembler". Reportlets support the visual assessment of intermediate preprocessing steps and final preprocessed outcomes, enabling researchers to evaluate the acceptability of the results efficiently. The assembler combines reportlets into a comprehensive document (a *final* visual report), providing a coherent and interpretable overview of the preprocessing workflow and the outcomes. *NiReports* not only provides the infrastructure to establish QA/QC protocols. By shedding light on the workflow operation itself, the visual reports provide "a scaffold for knowledge" that helps researchers better understand the why and the how of the particular operations in the workflow. This "educational" or transparency component supports the training and development of researchers, ultimately fostering a more knowledgeable and skilled community engaged in standardized preprocessing practices. Although *NiReports* offers some interactivity, standardization of QA/QC requires that the screening experience during assessments is homogenous across raters, regardless of their expertise or attrition or the actual visualization settings. In other words, *NiPreps*' reports disallow exploring data freely such that structured differences between raters may emerge depending on their strategy for the assessment. For instance, the reports do not

offer interactive ortho-viewers that permit two experts to navigate the same image differently. Standardizing QA/QC through *NiReports* ensures that preprocessing outcomes can be thoroughly and consistently assessed, providing researchers with confidence in the acceptability and quality of their results.

3.3 "Semantic" Standardization: Spatially Referencing Group Inferences

Standard spaces provide stereotaxy, a reference frame for researchers to align and compare data across different subjects and studies. Moreover, these spaces are ubiquitously employed in neuroimaging to incorporate prior knowledge into the processing as they typically are annotated. Standard spaces contain one or more *templates*, which are aggregated maps of neuroimaging features, and *atlases*—annotations corresponding to features encoded by templates in the frame of reference these engender. *TemplateFlow* [41] provides a curated collection of common standard templates and atlases, enabling self-adaptable workflows that employ the template or atlas most appropriate for the particular dataset. For instance, *fMRIPrep-infants* uses different templates depending on the age months of the participant. Although initially developed in response to increased flexibility requirements by *fMRIPrep*, we identified remarkable issues concerning the use and reporting of templates in neuroimaging as described in our *TemplateFlow* manuscript [41] and further analyzed in our feature about that paper [48]. Some of these issues relate to the distribution of templates and atlases under FAIR (Findability, Accessibility, Interoperability, and Reusability; [49]) guiding principles, to data management, as well as to ensuring best practices in reporting analyses. Therefore, not only does *TemplateFlow* offer a programmatic interface to templates and atlases and a community registry and archive, but it also addresses the challenge of template versioning and management (Fig. 4). By providing a centralized repository, researchers can access different versions of templates, allowing for consistent analyses across time and ensuring the reproducibility of preprocessing results.

3.4 Containerization

With growing requirements, it is harder for inexperienced users to adopt new tools because installation becomes a high barrier. These entry barriers increase with security policies and limitations of the target system. While installation on a Personal Computer (PC) may be straightforward, deployment on a multi-tenant HPC cluster can be challenging. To resolve this problem, *BIDS Apps* emphasized the adoption of containers (*see* Chapter 3). Indeed, when *fMRIPrep* started to ramp up in the number of users, most of the problems reported in the source code repository and the specialized forum *NeuroStars* (https://neurostars.org; *see* Resources) related to installation. As maintainers of *fMRIPrep*, we decided to discourage "bare-metal" installations where users must install all the dependencies (e.g., *AFNI*, *ANTs*, *FSL*, *FreeSurfer*, etc.) and prepare a workable *Python* environment themselves. Instead, we promoted

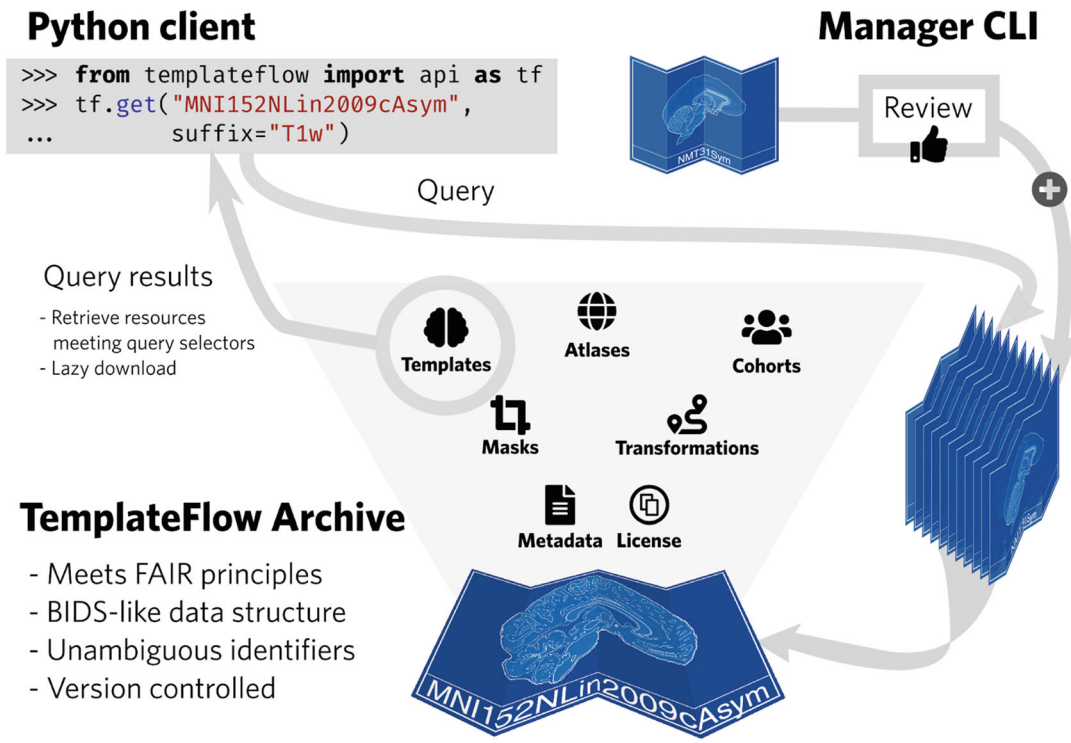

Fig. 4 Standardization of templates and atlases with *TemplateFlow*. The *TemplateFlow Archive* can be accessed at a "low" level with *DataLad*, or at a "high" level with the *TemplateFlow Client*. New resources can be added through the *TemplateFlow Manager* command-line interface, which initiates a peer-review process before acceptance in the *Archive*

the deployment of *Apptainer* (called *Singularity* at the time; [50]). We also successfully prompted the adoption of Singularity by our local cluster Sherlock (SCRR, Stanford University, CA, USA), and several systems at TACC (Texas Advanced Computing Center, University of Texas at Austin, TX, USA). The promotion of containers quickly translated into a shift of support activities toward more "scientific" topics about *fMRIPrep*, failure conditions, and feature requests, instead of installation and deployment (*see* Note).

Note: Running *fMRIPrep* with *Docker*
We build on top of the standard command line interface of *BIDS Apps* to demonstrate the containerized execution of *fMRIPrep* on the AOMIC-PIOP2 dataset [51].

```
docker run -ti --rm -u $( id -u ):$( id -g ) \
 -v /data/ds002790:/data:ro \
 -v /data/ds002790/derivatives:/derivatives \
```

(continued)

```
nipreps/fmriprep:23.2.0 \
/data /derivatives/fmriprep_23.2.0 participant \
--participant-label 0021 \
--omp-nthreads 8 --nprocs 16 \
-vv --bids-database-dir /data/.bids-index/
```

Execution with *Docker* requires pre-pending the container system's arguments before *fMRIPrep*'s. First, a `docker run` sub-command indicates that a container will be executed from a *Docker* image. The specific *Docker* image and tag marks the separation between *Docker* arguments and those of *fMRIPrep*. In this case, the fourth line indicates `nipreps/fmriprep:23.2.0`, thereby instructing *Docker* to find the corresponding image at the indicated version (23.2.0) and download it if not cached locally.

Arguments preceding the container image configure the terminal mode (`-ti`), instruct *Docker* to clear up the container when execution finishes (`--rm`), and, importantly, map the current user and group into the container (`-u $(id -u):$(id -g)`) to ensure new folders and files are not assigned to root, which is the default. Execution will also require file system communication with the container, and therefore, we "mount" the data folder in read-only mode (`-v /data/ds002790:/data:ro`), and a folder to store the output in read-write mode (`-v /data/ds002790/derivatives:/derivatives`).

Arguments to the right of the container name and tag correspond to *fMRIPrep*. First, we encounter the standard *BIDS Apps* mandatory arguments (`/data /derivatives/fmriprep_23.2.0 participant`), followed by one specific participant label (`--participant-label 0021`), as recommended in *fMRIPrep*'s usage guidelines. Next, parallelization is configured, with 8 CPUs per process and a maximum of 16 processes being executed simultaneously (`--omp-nthreads 8 --nprocs 16`). The verbosity of *fMRIPrep* can be tuned with the repetition of the "v" letter as a flag (here we have `-vv`, which can be decreased by writing just one "v", `-v`, or increased, e.g., `-vvvv`). This verbosity parameter should not be confused with *Docker*'s file system mounting flag `-v`. Finally, we set *PyBIDS*' cache by typing `--bids-database-dir /data/.bids-index/` (note how the directory is now relative to `/data`, the mount point inside the container where the dataset root will be available).

When starting with *fMRIPrep*, it is common to require several iterations to test configurations and arguments. If we

(continued)

are interested in keeping the intermediate results to speed up later executions of the pipeline, we need also to mount some (preferably fast) filesystem where these interim results are stored:

```
docker run -ti --rm -u $( id -u ):$( id -g ) \
 -v /data/ds002790:/data:ro \
 -v /data/ds002790/derivatives:/derivatives \
 -v /scratch/ds002790/sub-0021:/work \
 nipreps/fmriprep:23.2.0 \
 /data /derivatives/fmriprep_23.2.0 participant \
 --participant-label 0021 \
 --omp-nthreads 8 --nprocs 16 \
 -vv --bids-database-dir /data/.bids-index/ \
 --work /work
```

where `/scratch/ds002790/sub-0021` is a folder under a fast file system accessible at `/scratch`, and made accessible by the container as `/work`.

3.5 Telemetry

Telemetry enables the collection and analysis of data on the execution of processing pipelines, providing valuable insights into failure conditions and usage patterns (Figure 5 presents *fMRIPrep*'s telemetry). By incorporating telemetry, the *NiPreps* maintainers

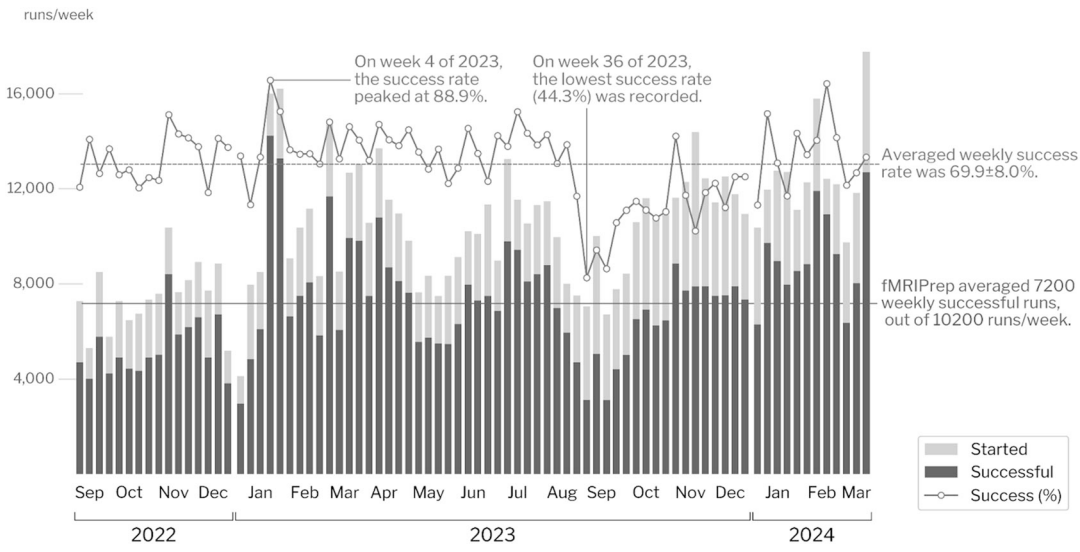

Fig. 5 *fMRIPrep* is executed an average of 11,200 times a week. By inserting telemetry instrumentation within *fMRIPrep*, we collect information to identify failure modes of the software and performance analytics. Over the past 1.9 years, *fMRIPrep* averages about 68% success rate

monitor their workflows' performance, identify potential bottlenecks and error modes, and optimize the pipeline accordingly. This information becomes invaluable in improving the reliability and efficiency of the pipeline process as it enables a deeper understanding of usage patterns by different users and provides unique insight into tracking actual software utilization [52]. Software impact metrics are a challenge to evaluate for open-source projects in general; however, these metrics are becoming relevant to funding bodies who value code as a scientific outcome. Initially, we implemented telemetry within *fMRIPrep* employing *Sentry* (https://sentry.io; *see* Resources). In order to generalize the analytics easily over the remainder of the *NiPreps* framework, we then developed *Migas* ("breadcrumbs" in Spanish) as an in-house solution for performance monitoring. *Migas* comprises a lightweight *Python* client that adds the necessary instrumentation to probe the application and submits (except when the user opts out by using the prescribed command line flag) the collected data to an Internet service running on the cloud.

3.6 Software Versioning and Release Cycle

Version control, code quality checking, testing, and continuous integration Version control and other software engineering best practices are crucial in achieving reliable and maintainable standardized workflows. Incorporating these practices is an onerous investment that will prevent scientific projects from incurring unsustainable technical debt quickly. Version control of scientific code is fundamental for reproducibility and traceability. By utilizing version control systems such as *Git*, researchers can track and manage changes to the workflow implementation over time (*see* Chapter 5 for *Git* use). It supports the management of different branches targeting specific features or release maintenance. By layering services over *Git*, research code developers can easily deploy quality checks (e.g., peer-review of code, use of "linters" to normalize the style of code and maximize collaboration, etc.), unit testing (at least, of clerical tasks such as filtering the input data structure, testing the accessibility of data and metadata, etc.), and continuous integration and continuous delivery (CI/CD). The *NiPreps*' documentation website[2] describes these techniques in further detail. Using *DataLad* (*see* Chapter 2 for a detailed description), *NiPreps* integrates "benchmarking" data in their CI/CD builds, automating the process of evaluating the acceptability of code changes and new features.

Versioned releases One early success driver for *fMRIPrep* was adopting a "release early, release often" or "RERO" release cycle, in which we would roll out new features and bug fixes rapidly (sometimes several times a week). This allowed *fMRIPrep* to stress

[2] https://www.nipreps.org/

test the implementation of new features and fixes in a federated fashion without the maintainers having access to the data. RERO was also appreciated by users, who identified the *fMRIPrep* developers as a responsive and supportive team, increasing the confidence that the tool would add reliability to their neuroimaging workflow. However, RERO requires that every new release is effectively identified by users, typically with a version label. We initially adopted semantic versioning [53] to assign these release identifiers, in which three numbers separated by periods (e.g., 2.1.10) contribute to the version interpretation. The first number or "major" differentiates hallmark iterations of the product. A "2.1.10" version indicates that a 1.x series of versions exists, and the technical gap of going from 1.x to 2.x is remarkable (e.g., outputs are incompatible, fundamental new features have been added, etc.). An example of such a large change was *Python*'s shift from 2.x into the current 3.x series. The second number, or "minor", signals large changes that are consistent enough to be considered under the same major (e.g., from *Python* 3.10 to 3.11). The last number, or "patch release", indicates small changes to address bugs or improve performance in a limited way. Starting in 2020, *fMRIPrep* and other *NiPreps* adopted a slightly different convention called calendar versioning,[3] which is fundamentally similar, but the major version number is replaced by the year of release. As a result, versions can be easier to place in the lifecycle of the software (*see* Fig. 6). For example, the current *fMRIPrep*'s last release is 23.2.0.

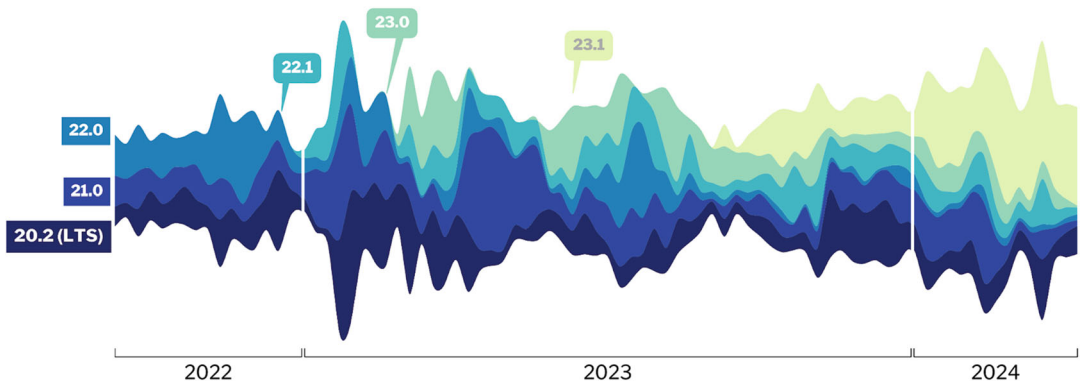

Fig. 6 Versioning and telemetry allow tracking the adoption of *fMRIPrep*. Adopting a strict versioning scheme for release permits monitoring the lifecycle of releases. With the riverplot below, we can identify many users who keep running the LTS (long-term support) version 20.2. Version 21.0 maintains many users, while later releases (22.1 and 23.0) progressively seem absorbed by the latest official release 23.1. This riverplot suggests users employing 21.0 series resist either falling back to the LTS or updating to the latest release

[3] https://www.nipreps.org/devs/releases/

Long-term support programs In addition to assigning meaningful version strings to every release, some of *fMRIPrep*'s power users expressed their concerns about our RERO approach in the context of longitudinal studies where data acquired during lengthy project spans require consistent processing throughout. As a solution, we established a "long-term support" (LTS) program[4,5] inspired by the *Ubuntu Linux* LTS program. Starting with version 20.2.0 (released on September 28, 2020), *fMRIPrep 20.2.x* is the active LTS series. LTS involves maintaining the software for a much longer window (preferably supported by researchers other than active maintainers and developers of *fMRIPrep*[6]), ensuring the continuity of the results over time, and resolving bugs discovered as the series are used. *fMRIPrep*'s LTS has seen seven "patch" updates; the last release is 20.2.7 on January 24, 2022 (please note how the major and minor numbers are pinned to the 20.2 series).

3.7 Community Involvement

Open scientific code is likely to egress the walls of the laboratory or research infrastructure that conceived it when the original authors and a community of researchers share the need and an ethos. Creating a community potentially ensures the project's longevity if it successfully engages researchers who follow up on the development of the tool and eventually contribute to it. "Contribution" here takes a broader meaning, as it comprehends not just code but also participation in discussions, the definition of roadmaps, providing support, writing documentation, etc. Second, nurturing a community helps access a more diverse pool of researchers reaching underrepresented and underserved minorities, which in the long term ensures the project does not decline due to monolithic thinking. For instance, the lack of researchers outside a given laboratory or consortium likely results in an inability to adapt to transformative advances elsewhere in the globe. Establishing standard procedures to make decisions and to keep communications fluent is necessary to ensure that the previous dimensions of standardization operate properly. For example, *fMRIPrep* and the newborn *NiPreps* initiated a process of creating a community around the framework in 2020. As a result, in April 2023, the *NiPreps* Governance (https://github.com/nipreps/GOVERNANCE) was passed by the community, and in September 2023, the first new "*NiPreps* Steering Committee" was selected. This process is not distinctive of the *NiPreps Community*, and indeed, our charter derives from a GitHub resource called "the minimally viable governance" or MVG.[7]

[4] https://www.nipreps.org/devs/releases/#long-term-support-series

[5] https://reproducibility.stanford.edu/fmriprep-lts/

[6] We thank Prof. Pierre Bellec and Dr. Basile Pinsard (CRIUGM, Psychology, University of Montreal) for the maintenance of *fMRIPrep 20.2.x LTS*.

[7] https://github.com/github/MVG

4 Challenges and Outlook

4.1 Challenges and Their Impact on Reproducibility

Probably, the number one question by neuroimaging experts that *fMRIPrep* elicited in its early days was, "What are the criteria for choosing one method over the alternatives at any given step?" Indeed, this query identifies a ubiquitous challenge due to limited objective evidence to compare the alternatives for each individual step and further the lack of combinatorial evidence when exploring the multiverse of tools. *fMRIPrep*'s choices have often relied on a combination of empirical findings, theoretical considerations, and expert opinions. The lack of programmatic, unambiguous, and comprehensive evidence for each preprocessing step makes the decision-making process challenging. It requires careful consideration of the trade-offs between different approaches, such as speed versus accuracy, robustness versus sensitivity, and generalizability versus specificity. Moreover, even if there was clear evidence to drive these choices, it is likely that different objective functions will yield different "best" options. Indeed, we compared several fMRI preprocessing workflows [54] by implementing them within the *Configurable Pipeline for the Analysis of Connectomes* (*C-PAC*; [55]). Although these implementations did not replicate the exact workflows, the results highlighted significant variations across different preprocessing approaches. While some convergences in functional connectivity results were observed, the overall variability demonstrated the challenges of obtaining consistent outcomes across different implementations.

Relatedly, the development of *fMRIPrep* has often generated discussions about the balance between enabling many options for the user and how that extra analytical flexibility may undermine the reproducibility of the analyses. Standardization removes a researcher's degrees of freedom by making choices (e.g., selecting a brain extraction algorithm that fails in one image per million but is less accurate than an alternative approach with a failure rate of ten images per million and extremely precise otherwise), and by adding friction points (e.g., conversion into *BIDS*) that require the researcher to be aware of, and explicit about, all the experimental details.

Since analytical variability is only one factor in the overall reliability, it is critical to understand and account for the variability introduced by each specific step in the preprocessing pipeline. In collaboration with the CRIUGM team at the University of Montreal, we tested the reliability of *fMRIPrep* by introducing small random numerical variabilities at some anatomical preprocessing steps with *libmath* [56]. This approach allowed for the assessment of how uncertainties propagate throughout the pipeline. The paper proposed a method to identify large discontinuities between different versions in the development cycle. Indeed we identified

implementation changes between two consecutive "patch" releases (20.2.4 to 20.2.5) that introduced large changes—hence violating the principle that patch version increments should be backward compatible. Therefore, the approach provides valuable insights into the stability and reproducibility of the pipeline over different versions, aiding in detecting potential sources of variability (*see* Fig. 7 in ref. [56]).

Establishing QC criteria with reference to the specific application requires the definition standards such as quality metrics applicable to pipeline outcomes. For example, in Fig. 3A of ref. [24], we compare the outputs *fMRIPrep* and *FSL FEAT* in terms of data smoothness. Smoothness is likely a quality metric of interest to high-resolution BOLD data acquired with 7 Tesla devices, as they often showcase excessive smoothing after processing due to, e.g., multiple resamplings.

Further challenges relate to resource utilization and their optimization. Optimizing resource utilization for specific solutions is markedly easier than with standardized alternatives. As a quick example, while *PyBIDS* takes a few seconds to index a *BIDS* directory for a small-sized dataset (15 subjects including one session each with minimal anatomical data and some diffusion or functional MRI), it may take one hour on the same computer for a dataset of 1500 subjects and similar imaging contents per subject. Further, a custom workflow developed for a specific sample of 100 neurotypical subjects of a narrow age range, collected on a single scanner and with a single imaging protocol, will be uniform in imaging parameters (e.g., size, resolution, contrast, artifacts, etc.). Therefore, all subjects will have similar demands from the compute resource regarding memory. However, anticipating memory requirements for an equivalent standard workflow that is expected to perform properly on diverse samples in terms of both phenotypes and imaging parameters is challenging, as images may come in many different sizes (e.g., some have very large acquisition matrices) and other fundamental properties (e.g., anisotropic vs. isotropic voxels, healthy vs. lesioned brains, etc.). Resource utilization challenges are further constrained by the need for and responsibility of reducing the carbon footprint of executing these pipelines [57].

4.2 The Need for Openly Available and Reusable Data

The availability of diverse fMRI data, accessible under FAIR principles [49] and readily reusable thanks to *BIDS*, was critical to developing *fMRIPrep* and *MRIQC*. To develop *fMRIPrep*, we leveraged a long list of datasets available at the time through *OpenfMRI* [58] and *OpenNeuro* [22]. In the case of *MRIQC*, we employed two specific datasets: ABIDE [59] and the Consortium for Neuropsychiatric Phenomics dataset [60]. Open data will remain essential to any methodological development endeavor to address the question, "Does this software work on a substantial

number of diverse studies?" In [24], we visually assessed performance on 54 datasets using the standard reports. Further, new datasets tailored to methodological development, such as our "Human Connectome Phantom" (HCPh; [34]), will be necessary to face the need to explore the multiverse of methodological choices. More importantly, open data will be necessary to face the new challenges derived from adopting artificial intelligence (Subheading 4.3). While deep learning algorithms often exhibit great performance, they largely operate as "opaque boxes", which impedes checking how inference was made. This lack of interpretability limits our understanding of the underlying factors driving the model's outputs. Open data is fundamental to validate findings and compare results across different datasets.

4.3 Future Directions

The advent of artificial intelligence (AI) and deep learning The introduction of emerging deep learning models into standardized processing pipelines faces friction, as the "classical" computer vision techniques have been more thoroughly tested and have engendered trust in their performance. Conversely, the complexity of deep learning models raises concerns about transparency and interpretability. Nonetheless, deep-learning applications have demonstrated great reliability on diverse data and remarkably better performance, drastically reducing inference times while improving robustness. One example of such a transition currently being tested within *NiPreps* is *SynthStrip* [61], a multi-modal human brain extraction tool. The *FreeSurfer* team is at the forefront of leveraging deep learning in the processing pipeline and has introduced a range of relevant tools, including *SynthStrip*, *SynthSeg* [62], or *SynthSR* [63]. These particular tools have the common denominator of being designed to achieve great reliability independently of the image modality, a requirement perfectly aligned with the cross-modality standardization goals of *NiPreps*.

Fully differentiable pipelines We have also demonstrated the potential of a fully differentiable software stack in the domain of functional connectomics [64]. We argue that differentiable programming does not resolve the problem of workflow design but rather is a tool to free workflow design from the analytical choices and convert it in a hyperparameter search process. Hence, the challenge of methodological choices is resolved by data-driven optimization. Ciric's *hypercoil* software (https://hypercoil.github.io) is based on *PyTorch* and enables end-to-end differentiability throughout the entire preprocessing pipeline. In connection with the challenge of making decisions about the particular implementation of each processing step, a fully differentiable pipeline resolves the problem in a data-driven way with an objective function built in. As a result, the user is, in principle, not offered knobs (degrees of freedom) to tune the processing.

AI-assisted code writing As Poldrack and colleagues contend [65], code assisting is one of the tasks where large language models particularly shine. Even if the aid is limited to "only revising" a given code, they encountered substantial improvement in the code by several metrics. However, they found some limitations in generating tests, suggesting that humans are still necessary to ensure the validity and accuracy of the results. Nonetheless, their findings point to AI as the key to multiplying the productivity of humans by perfecting many of the "almost-clerical" tasks that standardization requires.

5 Conclusion

This chapter discusses the rationale, benefits, and challenges of standardizing preprocessing in neuroimaging. We have presented *NiPreps*, a modular and adaptable workflow framework that aims to provide reliable and reproducible preprocessing solutions for different modalities, populations, and species. We have also described some of the best practices and tools *NiPreps* employs to ensure the preprocessing results' quality, consistency, and transparency, such as *BIDS*, *TemplateFlow*, *NiReports*, *Migas*, containerization, and version control. We have highlighted some of the current and future directions of *NiPreps*, such as incorporating deep learning models, enabling end-to-end differentiability, and leveraging AI-assisted code writing. We have also emphasized the importance of open and reusable data for validating and comparing different preprocessing approaches and enhancing the interpretability and generalizability of the outcomes.

We hope that this chapter has provided a comprehensive overview of the state of the art and challenges of standardizing preprocessing in neuroimaging. We believe *NiPreps* offers a valuable resource for researchers seeking to optimize their preprocessing pipelines and obtain high-quality and robust results. We invite the readers to join the *NiPreps* community and contribute to the development and improvement of the framework. By adopting and promoting standardized preprocessing practices, we can advance the field of neuroimaging and foster a more collaborative and reproducible scientific culture.

Acknowledgments

The author and this work are supported by the Swiss National Science Foundation—SNSF—(#185872), NIMH (RF1MH12186), and CZI (EOSS5-0000000266).

Thanks to M.Sc. Céline Provins, Dr. Vanessa Siffredi, Dr. Martin Nørgaard, Dr. Yasser Alemán-Gómez, Dr. Jon Haitz Legarreta Gorroño, Dr. Satrajit S. Ghosh, and Prof. Russell A. Poldrack for their invaluable feedback along iterations of the draft and their generous scientific support in our collaborations. Thanks to the editors (Dr. Hervé Lemaître and Dr. Robert Whelan) for their infinite patience and support when developing this chapter. Thanks to the *NiPreps Community* for their continued support and their energy in pursuit of a more transparent and reliable neuroimaging.

References

1. Strother SC (2006) Evaluating fMRI preprocessing pipelines. IEEE Eng Med Biol Mag 25(2):27–41. https://doi.org/10.1109/MEMB.2006.1607667
2. Cox RW, Hyde JS (1997) Software tools for analysis and visualization of fMRI data. NMR Biomed 10(4–5):171–178. https://doi.org/10.1002/(SICI)1099-1492(199706/08)10:4/5<171::AID-NBM453>3.0.CO;2-L
3. Fischl B (2012) FreeSurfer. NeuroImage 62(2):774–781. https://doi.org/10.1016/j.neuroimage.2012.01.021
4. Jenkinson M, Beckmann CF, Behrens TEJ, Woolrich MW, Smith SM (2012) FSL. NeuroImage 62(2):782–790. https://doi.org/10.1016/j.neuroimage.2011.09.015
5. Friston KJ, Ashburner J, Kiebel SJ, Nichols TE, Penny WD (2006) Statistical parametric mapping : the analysis of functional brain images. Academic Press, London
6. Brett M et al (2006) Open source software: NiBabel. Zenodo Softw:3458246. https://doi.org/10.5281/zenodo.591597
7. Gorgolewski KJ et al (2011) Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in Python. Front Neuroinform 5:13. https://doi.org/10.3389/fninf.2011.00013
8. Carp J (2012) On the plurality of (methodological) worlds: estimating the analytic flexibility of fMRI experiments. Front Neurosci 6. https://doi.org/10.3389/fnins.2012.00149
9. Bowring A, Maumet C, Nichols TE (2019) Exploring the impact of analysis software on task fMRI results. Hum Brain Mapp (in press). https://doi.org/10.1002/hbm.24603
10. Bowring A, Nichols TE, Maumet C (2022) Isolating the sources of pipeline-variability in group-level task-fMRI results. Hum Brain Mapp 43(3):1112–1128. https://doi.org/10.1002/hbm.25713
11. Nørgaard M et al (2020) Different preprocessing strategies lead to different conclusions: a [11C]DASB-PET reproducibility study. J Cereb Blood Flow Metab 40(9):1902–1911. https://doi.org/10.1177/0271678X19880450
12. Botvinik-Nezer R et al (2020) Variability in the analysis of a single neuroimaging dataset by many teams. Nature 582(7810):7810. https://doi.org/10.1038/s41586-020-2314-9
13. Churchill NW, Spring R, Afshin-Pour B, Dong F, Strother SC (2015) An automated, adaptive framework for optimizing preprocessing pipelines in task-based functional MRI. PLoS One 10(7):e0131520. https://doi.org/10.1371/journal.pone.0131520
14. Dafflon J et al (2022) A guided multiverse study of neuroimaging analyses. Nat Commun 13(1):1. https://doi.org/10.1038/s41467-022-31347-8
15. Allen C, Mehler DMA (2019) Open science challenges, benefits and tips in early career and beyond. PLoS Biol 17(5):e3000246. https://doi.org/10.1371/journal.pbio.3000246
16. Chambers CD, Tzavella L (2022) The past, present and future of registered reports. Nat Hum Behav 6(1):1. https://doi.org/10.1038/s41562-021-01193-7
17. Ozenne B, Norgaard M, Pernet C, Ganz M (2024) A sensitivity analysis to quantify the impact of neuroimaging preprocessing strategies on subsequent statistical analyses. Apr. 24, 2024, *arXiv*: arXiv:2404.14882. https://doi.org/10.48550/arXiv.2404.14882
18. Nichols TE et al (2017) Best practices in data analysis and sharing in neuroimaging using MRI. Nat Neurosci 20:299–303. https://doi.org/10.1038/nn.4500

19. Taylor PA et al (2023) Highlight results, don't hide them: enhance interpretation, reduce biases and improve reproducibility. NeuroImage 274:120138. https://doi.org/10.1016/j.neuroimage.2023.120138
20. Gorgolewski KJ et al (2016) The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Sci Data 3:160044. https://doi.org/10.1038/sdata.2016.44
21. Poldrack RA et al (2024) The past, present, and future of the brain imaging data structure (BIDS). Imaging Neurosci 2:1–19. https://doi.org/10.1162/imag_a_00103
22. Markiewicz CJ et al (2021) The OpenNeuro resource for sharing of neuroscience data. eLife 10:e71774. https://doi.org/10.7554/eLife.71774
23. Gorgolewski KJ et al (2017) BIDS apps: improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. PLoS Comput Biol 13(3):e1005209. https://doi.org/10.1371/journal.pcbi.1005209
24. Esteban O et al (2019) fMRIPrep: a robust preprocessing pipeline for functional MRI. Nat Methods 16(1):111–116. https://doi.org/10.1038/s41592-018-0235-4
25. Esteban O et al (2020) Analysis of task-based functional MRI data preprocessed with fMRIPrep. Nat Protoc 15:2186–2202. https://doi.org/10.1101/694364
26. Niso G et al (2022) Open and reproducible neuroimaging: from study inception to publication. NeuroImage:119623. https://doi.org/10.1016/j.neuroimage.2022.119623
27. Novick MR (1966) The axioms and principal results of classical test theory. J Math Psychol 3(1):1–18. https://doi.org/10.1016/0022-2496(66)90002-2
28. Lord FM, Novick MR, Birnbaum A (1968) Statistical theories of mental test scores. In: Statistical theories of mental test scores. Addison-Wesley, Oxford, Uk
29. Allen MJ, Yen WM (1979) Introduction to measurement theory. Waveland Press
30. Plesser HE (2018) Reproducibility vs. replicability: a brief history of a confused terminology. Front Neuroinform 11. https://doi.org/10.3389/fninf.2017.00076
31. Brown WM (1910) Some experimental results in the correlation of mental abilities. Br J Psychol 3:296–322
32. Spearman C (1910) Correlation calculated from faulty data. Br J Psychol 3:271–295
33. Naselaris T, Allen E, Kay K (2021) Extensive sampling for complete models of individual brains. Curr Opin Behav Sci 40:45–51. https://doi.org/10.1016/j.cobeha.2020.12.008
34. Provins C et al (2023) Reliability characterization of MRI measurements for analyses of brain networks on a single human. Nat Methods (Stage 1 accepted-in-principle). https://doi.org/10.17605/OSF.IO/VAMQ6
35. Cronbach LJ, Meehl PE (1955) Construct validity in psychological tests. Psychol Bull 52(4):281–302. https://doi.org/10.1037/h0040957
36. Yarkoni T et al (2019) PyBIDS: python tools for BIDS datasets. J Open Source Softw 4:1294. https://doi.org/10.21105/joss.01294
37. Esteban O, Birman D, Schaer M, Koyejo OO, Poldrack RA, Gorgolewski KJ (2017) MRIQC: advancing the automatic prediction of image quality in MRI from unseen sites. PLoS One 12(9):e0184661. https://doi.org/10.1371/journal.pone.0184661
38. Pedregosa F et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
39. Markiewicz CJ et al (2024) FMRIPrep-next: preprocessing as a fit-transform model. In: Annual meeting of the Organization for Human Brain Mapping (OHBM), Seoul, Korea, p (accepted)
40. Goncalves M et al (2023) They grow up so fast – augmenting the NiBabies infant MRI workflow. In: Annual meeting of the Organization for Human Brain Mapping (OHBM), Montréal, Canada, p 2504
41. Ciric R et al (2022) TemplateFlow: FAIR-sharing of multi-scale, multi-species brain models. Nat Methods 19:1568–1571. https://doi.org/10.1038/s41592-022-01681-2
42. Esteban O et al (2021) The Bermuda Triangle of d- and f-MRI sailors – software for susceptibility distortions (SDCFlows). In: 27th Annual Meeting of the Organization for Human Brain Mapping, Virtual Meeting, p 1653. https://doi.org/10.31219/osf.io/gy8nt
43. MacNicol EE, Hagen MP, Provins C, Kim E, Cash D, Esteban O (2022) Extending MRIQC to rodents: image quality metrics for rat MRI. In: Annual meeting of the European Society for Molecular Imaging (EMIM), Thessaloniki, Greece, pp PW23–P913
44. Joseph MJE et al (2021) dMRIPrep: a robust preprocessing pipeline for diffusion MRI. In: Proceedings of the International Society for Magnetic Resonance in Medicine, Virtual Meeting, p 2473. Accessed: Mar. 12, 2021. [Online]. Available: https://docs.google.com/document/u/2/d/1ocamAFP2

OGnUIUooL9gxu5CExqiCS4Le3caHWM_8E04/edit?usp=drive_web&ouid=104994410621175933959&usp=embed_facebook

45. Adebimpe A et al (2022) ASLPrep: a generalizable platform for processing of arterial spin Labeled MRI and quantification of regional brain perfusion. Nat Methods 19:683–686. https://doi.org/10.1038/s41592-022-01458-7
46. Provins C, MacNicol EE, Seeley SH, Hagmann P, Esteban O (2023) Quality control in functional MRI studies with MRIQC and fMRIPrep. Front Neuroimaging 1:1073734. https://doi.org/10.3389/fnimg.2022.1073734
47. Reason J, Broadbent DE, Baddeley AD, Reason J (1997) The contribution of latent human failures to the breakdown of complex systems. Philos Trans R Soc Lond B Biol Sci 327(1241):475–484. https://doi.org/10.1098/rstb.1990.0090
48. (2022) Harnessing the multiverse of neuroimaging standard references. Nat Methods 19(12):12. https://doi.org/10.1038/s41592-022-01682-1
49. Wilkinson MD et al (2016) The FAIR guiding principles for scientific data management and stewardship. Sci Data 3(1):1. https://doi.org/10.1038/sdata.2016.18
50. Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: scientific containers for mobility of compute. PLoS One 12(5):e0177459. https://doi.org/10.1371/journal.pone.0177459
51. Snoek L, van der Miesen MM, Beemsterboer T, van der Leij A, Eigenhuis A, Steven Scholte H (2021) The Amsterdam open MRI collection, a set of multimodal MRI datasets for individual difference analyses. Sci Data 8(1):85. https://doi.org/10.1038/s41597-021-00870-6
52. Afiaz A et al (2023) Evaluation of software impact designed for biomedical research: Are we measuring what's meaningful? June 05, 2023, arXiv: arXiv:2306.03255. https://doi.org/10.48550/arXiv.2306.03255
53. Preston-Werner T (2024) Semantic Versioning 2.0.0, Semantic Versioning. Accessed: Feb 22, 2024. [Online]. Available: https://semver.org/
54. Li X et al (2024) Moving beyond processing and analysis-related variation in neuroscience. Nat Hum Behav (accepted):2021.12.01.470790. https://doi.org/10.1101/2021.12.01.470790
55. Sikka S et al (2014) Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (C-PAC). In: 5th INCF Congress of Neuroinformatics, Munich, Germany. https://doi.org/10.3389/conf.fninf.2014.08.00117
56. Chatelain Y et al (2023) A numerical variability approach to results stability tests and its application to neuroimaging, July 10, 2023, *arXiv*: arXiv:2307.01373. https://doi.org/10.48550/arXiv.2307.01373
57. Souter NE et al (2023) Ten recommendations for reducing the carbon footprint of research computing in human neuroimaging. Imaging Neurosci 1:1–15. https://doi.org/10.1162/imag_a_00043
58. Poldrack RA et al (2013) Toward open sharing of task-based fMRI data: the OpenfMRI project. Front Neuroinform 7:12. https://doi.org/10.3389/fninf.2013.00012
59. Di Martino A et al (2014) The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol Psychiatry 19(6):659–667. https://doi.org/10.1038/mp.2013.78
60. Poldrack RA et al (2016) A phenome-wide examination of neural and cognitive function. Sci Data 3:160110. https://doi.org/10.1038/sdata.2016.110
61. Hoopes A, Mora JS, Dalca AV, Fischl B, Hoffmann M (2022) SynthStrip: skull-stripping for any brain image. NeuroImage 260:119474. https://doi.org/10.1016/j.neuroimage.2022.119474
62. Billot B et al (2023) SynthSeg: segmentation of brain MRI scans of any contrast and resolution without retraining. Med Image Anal 86:102789. https://doi.org/10.1016/j.media.2023.102789
63. Iglesias JE et al (2023) SynthSR: a public AI tool to turn heterogeneous clinical brain scans into high-resolution T1-weighted images for 3D morphometry. Sci Adv 9(5):eadd3607. https://doi.org/10.1126/sciadv.add3607
64. Ciric R, Thomas AW, Esteban O, Poldrack RA (2022) Differentiable programming for functional connectomics. In: Proceedings of the 2nd Machine Learning for Health symposium. PMLR, New Orleans, pp 419–455. Accessed: Dec. 05, 2022. [Online]. Available: https://proceedings.mlr.press/v193/ciric22a.html
65. Poldrack RA, Lu T, Beguš G AI-assisted coding: experiments with GPT-4, Apr. 25, 2023, *arXiv*: arXiv:2304.13187. https://doi.org/10.48550/arXiv.2304.13187