

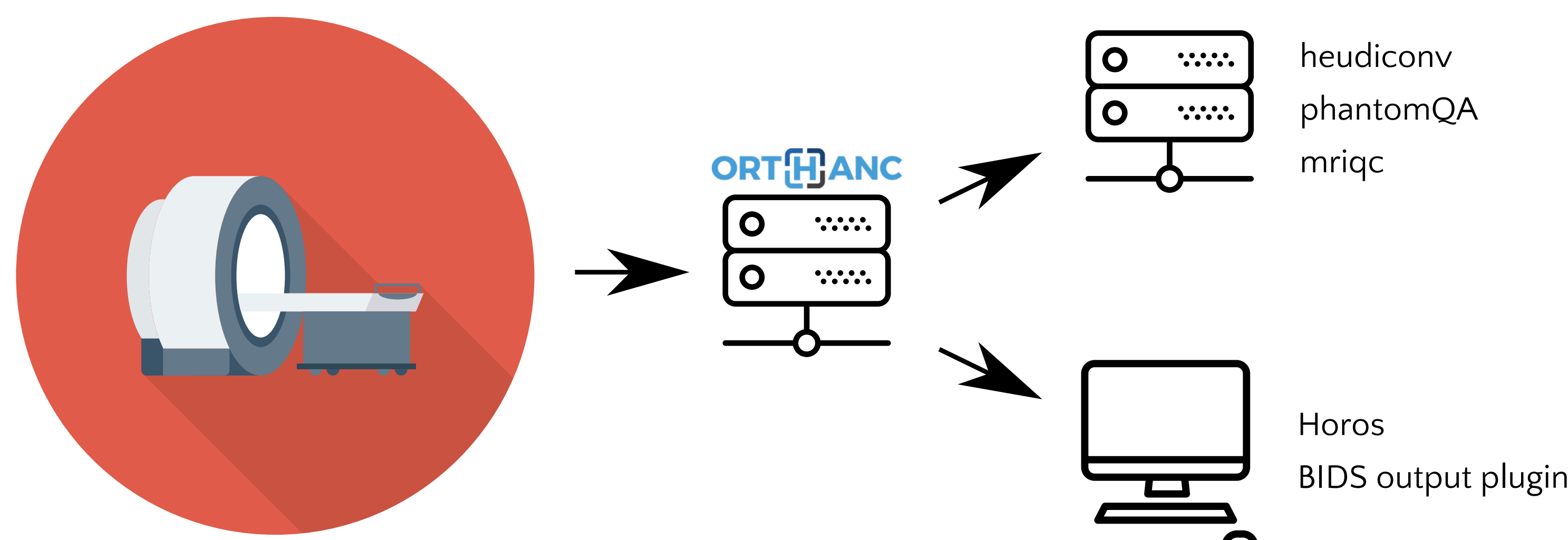
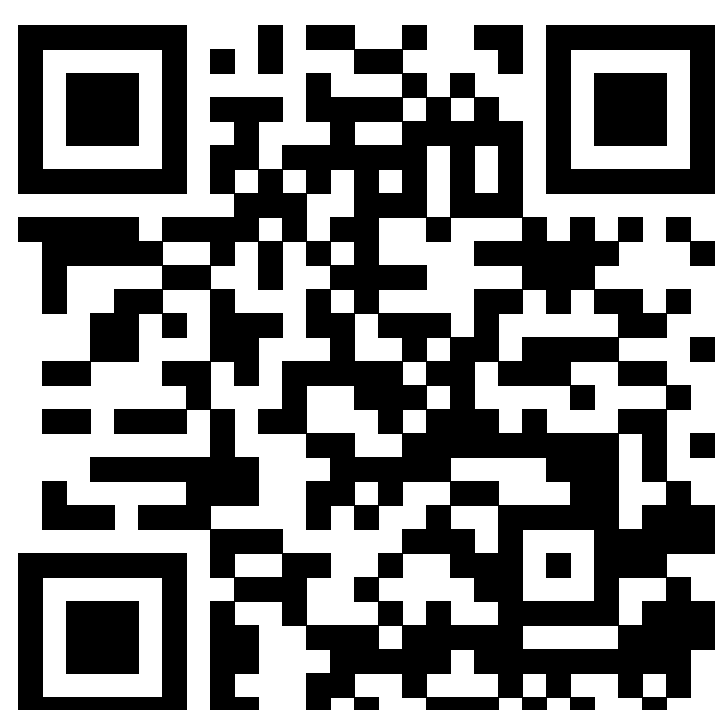
Data export plugin and automated pipeline based on BIDS. Expanding and utilising free software to improve fMRI data flow in Laboratory of Brain Imaging.

Motivation

Laboratory of Brain Imaging is a core facility at Nencki Institute, equipped with 3T MRI scanner available for both internal and external research groups. This means that in a typical month there are around **10 research projects** sharing the scanner time. The emergence of **Brain Imaging Data Structure**, a standard of data organisation, and related **free software ecosystem** prompted us to **explore ways of improving the data flow** from the scanner to the researchers. Here we describe two projects: a **data export plugin** and an **automated pipeline** which starts every time a scanning session is completed.

Online version

Mobile friendly, with PDF & links:
nencki-lobi.github.io/bids-flow/



What is BIDS

- Standard for organisation of brain imaging data & metadata (not a file format)
- Intuitive for users and machine-readable (see basic example below)
- Intended to enable automated tools with minimal input
- Initially for MRI, now also MEG, EEG and iEEG.
- See bids.neuroimaging.io

```
sub-Abcdef/  
  anat/  
    sub-Abcdef_T1w.nii.gz  
  func/  
    sub-Abcdef/task-nback_run-1_bold.nii.gz  
    sub-Abcdef/task-nback_run-1_bold.json  
    sub-Abcdef/task-nback_run-1_events.tsv
```

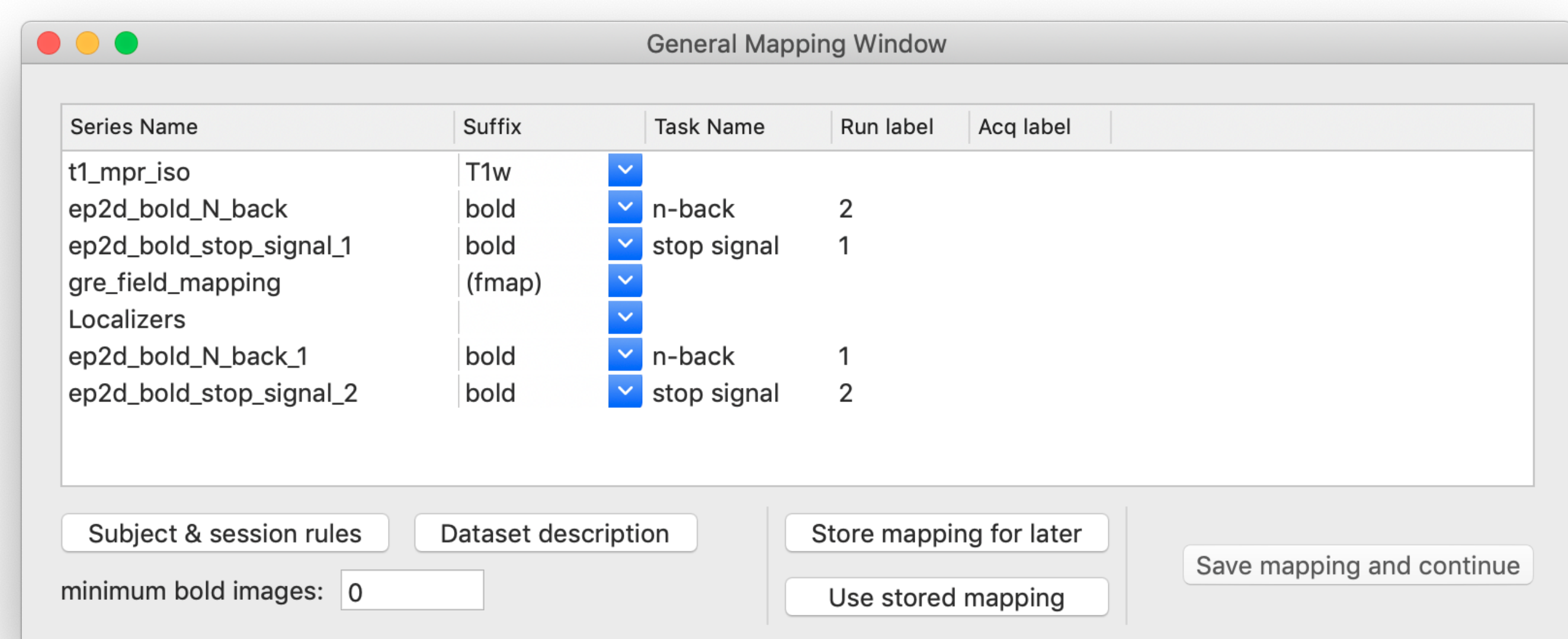
What are BIDS-Apps

- Container image capturing a neuroimaging pipeline which takes a BIDS formatted dataset as input
- Container: package of software with all dependencies
- Uses Docker or Singularity to run independent on OS (Linux / Windows / MacOS)
- Apps have the same core set of arguments, can run with minimal input (see below)
- See bids-apps.neuroimaging.io

```
docker run -it --rm          # docker and its options  
-v <bids_dir>:/data:ro      # mount data folder, read only  
-v <out_dir>:/out           # mount output folder  
bids/example:0.0.4         # specify app to be pulled from Docker Hub  
/data /out participant      # input path, output path, analysis level
```

Horos BIDS output plugin

We created a data export extension for Horos DICOM browser. According to our knowledge, this is the first GUI-based tool which can be used to automatically create BIDS datasets.



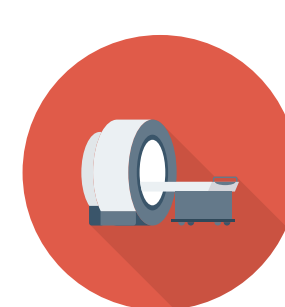
- **Graphical Interface** – friendly for non-technical users
- **Flexible** – few initial assumptions as detected sequences are annotated by users
- **GPL license** – available from github.com/mslw/horos-bids-output
- **Documented** – usage instructions and explanations are provided on github
- **Works within Horos** – powerful database and networking features, MacOS only
- Prior to exporting, Horos can fetch data from a remote server
- Plugin uses Horos API for DICOM queries
- Plugin manages file names and uses **dcm2niix** for conversion
- Simple heuristics for multiple runs or discarded scans
- Written in Objective-C using Cocoa framework

List of relevant software

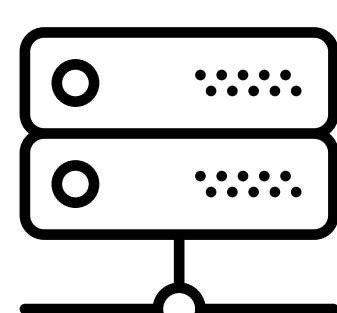
Name	URL	License
Horos plugin	github.com/mslw/horos-bids-output	GPL
BIDS-Flow	github.com/nencki-lobi/bids-flow	-
Horos	horosproject.org	LGPL
Dcm2niix	github.com/rordenlab/dcm2niix	BSD + PD + MIT
Orthanc	orthanc-server.com	GPL
Heudiconv	github.com/nipy/heudiconv	Apache
Mriqc	mriqc.readthedocs.io	BSD 3-Clause
Docker	docker.com	Apache
Fmriprep	fmriprep.readthedocs.io	BSD 3-Clause
Reproin	github.com/ReproNim/reproin	MIT
XNAT	xnat.org	own open source

Custom automated pipeline: BIDS-flow

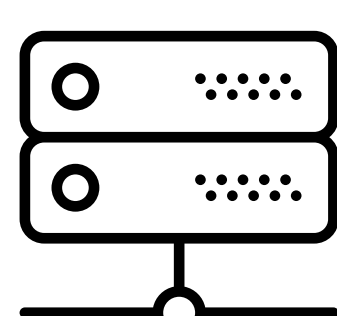
We combined a set of existing tools with lua, python and bash scripting to create an automated pipeline, beginning with raw data coming off the scanner and ending with quality control reports accessible for users. We were inspired by the ReproIn project, but took a learn-by-doing approach to build a system which is less generic but more suited to our current needs and capabilities. Our code is available on github.com/nencki-lobi/bids-flow.



An agreed upon set of sequence names
(task-taskname, T1w, phantom_, ...)



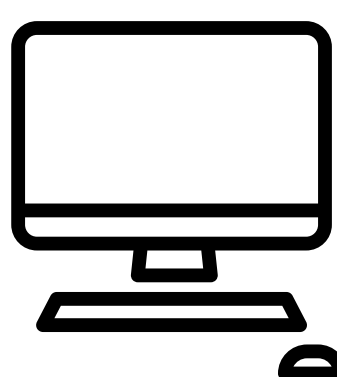
PACS (Picture Archiving and Communications System) server
Orthanc software to manage files and database
OnStableStudy lua script to copy files after each subject



Computational server (currently a repurposed PC)
Cron job: "sentinel.sh" script decides what to do next
Next steps in separate bash scripts, chosen based on sequence names

- For phantom, run fBIRN phantom QA
- For regular study, convert ro BIDS/NIfTI files using heudiconv (custom heuristic)
- Submit the BIDS data to mriqc BIDS app
- Any other BIDS-App could potentially be used in the same way

Both phantom and regular QC generate html reports, published on local network



Users can monitor recent scans and trends over time
Goal: raw data, QC reports and processed data all available to investigators

We are considering **further possibilities**: either **OS way** (manage user accounts, set file permission, allow user-specific configurations) or **moving to XNAT** platform (XNAT has a powerful set of tools for organising data, running pipelines remotely and managing data access through a web browser interface).

Bibliography

Gorgolewski, Krzysztof J., Tibor Auer, Vince D. Calhoun, R. Cameron Craddock, Samir Das, Eugene P. Duff, Guillaume Flandin, et al.

"The Brain Imaging Data Structure, a Format for Organizing and Describing Outputs of Neuroimaging Experiments."
Scientific Data 3 (June 21, 2016): 160044. <https://doi.org/10.1038/sdata.2016.44>.

Computer icon made by freepik, MRI icon made by Dinosoft Labs, Server icon made by Smashicons – from flaticon.com

Orthanc logo from Orthanc website