

Introduction

Neuroimaging is undergoing a methodological mutation with new standards for data organization (Brain Imaging Data Structure, or BIDS) [1] or command line description (Boutiques) [2]; and with the popularization of software containers technologies such as docker or singularity for easy software distribution.

Using these solutions, fully automatic processing pipelines can be easily shared and reused with standard commands and without further installation of software. BIDS apps (<https://github.com/BIDS-Apps>), for instance, propose a list of analysis pipelines, installed on Docker containers, that can be reused easily by any other group on a BIDS compatible dataset. One major limitation of BIDS app or similar solution is the lack of flexibility and transparency: understanding, adapting, modifying and creating new pipelines requires advanced skills that hinders the use of these solutions. [3] [4]

Here, we propose to use the very popular and stable SPM batch system to build flexible and transparent pipelines that can (1) call any dockerized neuroimaging tool (installed on a docker container), (2) parse a BIDS directory to perform individual or group analysis and (3) integrates a flexible quality control system. For this aim, new SPM modules have been developed. This open source solution can be used for free on any platform (Windows, Mac or Linux), using the sum apps provided at the following link:

<https://github.com/tanguyduval/spm12>

Conclusion

In this work, we propose a set of SPM apps for making accessible recent neuroimaging standards to the greatest number of researchers. These standards facilitate multi-centric studies and reproducible science. The solution relies on the popular and stable SPM batch system, and on the container technology Docker to get rid of painstaking software installation and dependencies. It conforms to recently growing standards such as BIDS and Boutiques.

References

- [1] K.J. Gorgolewski et al., Scientific data, 2016, 3, 160044.
- [2] T. Glatard et al., arXiv [cs.SE], 2017.
- [3] P. Bellec et al., Frontiers in neuroinformatics, 2012, 6, 7.
- [4] K. Gorgolewski et al., Frontiers in neuroinformatics, 2011, 5, 13.
- [5] <https://www.mathworks.com/matlabcentral/fileexchange/42997-xiangruili-dcm2nii>
- [6] B. De Leener et al., NeuroImage, 2017, 145, 24.

BIDS app

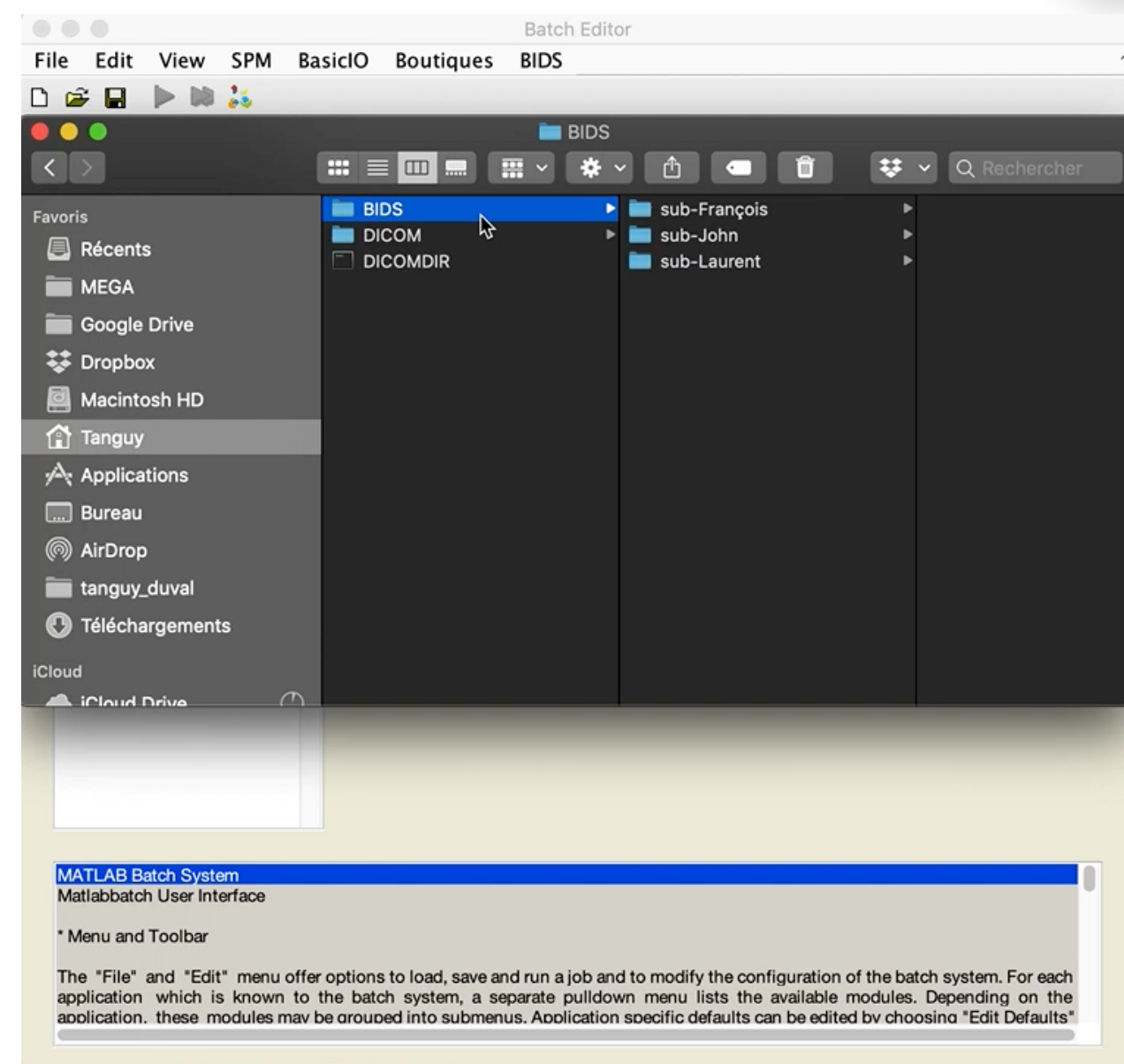
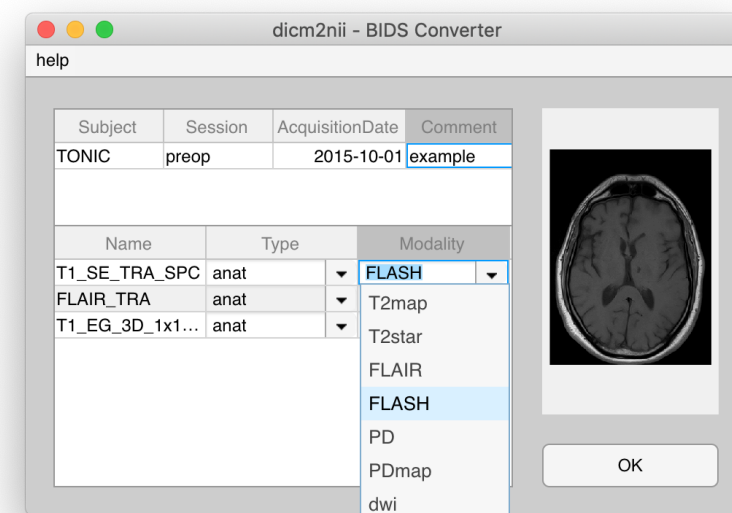
BIDS is the standard for neuroimaging data organization: <http://bids.neuroimaging.io>

► This figure shows the **Interactive Dicom to BIDS converter**

The converter is based on the popular and reliable dcm2bids converter [5] and provides a simple graphical user interface with slice preview and a memory that learns to recognize serie description (is it a T1w? a dwi?).

▼ This video is a demo of the **BIDS folder parser**

For running pipelines on a selection of subjects and for creating and sharing generic pipelines compatible with any BIDS-compatible dataset.

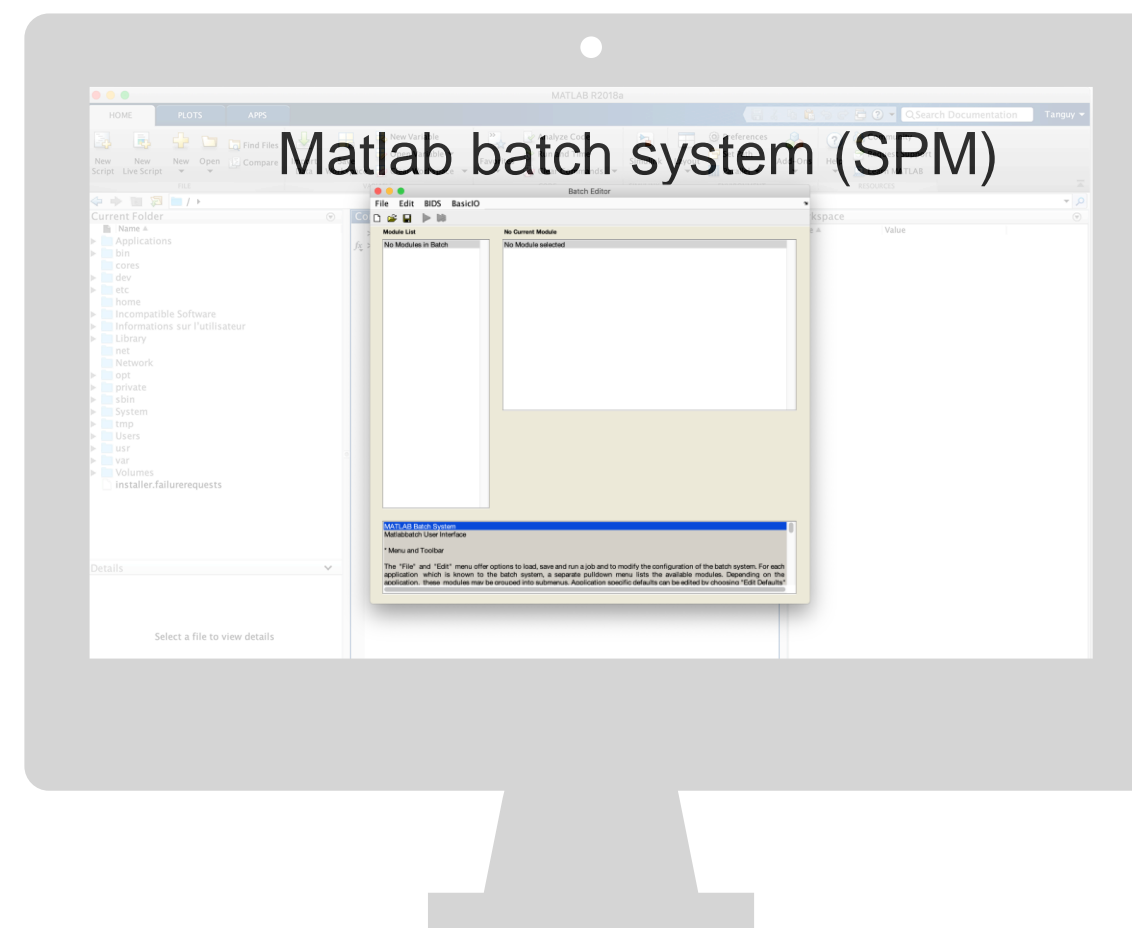
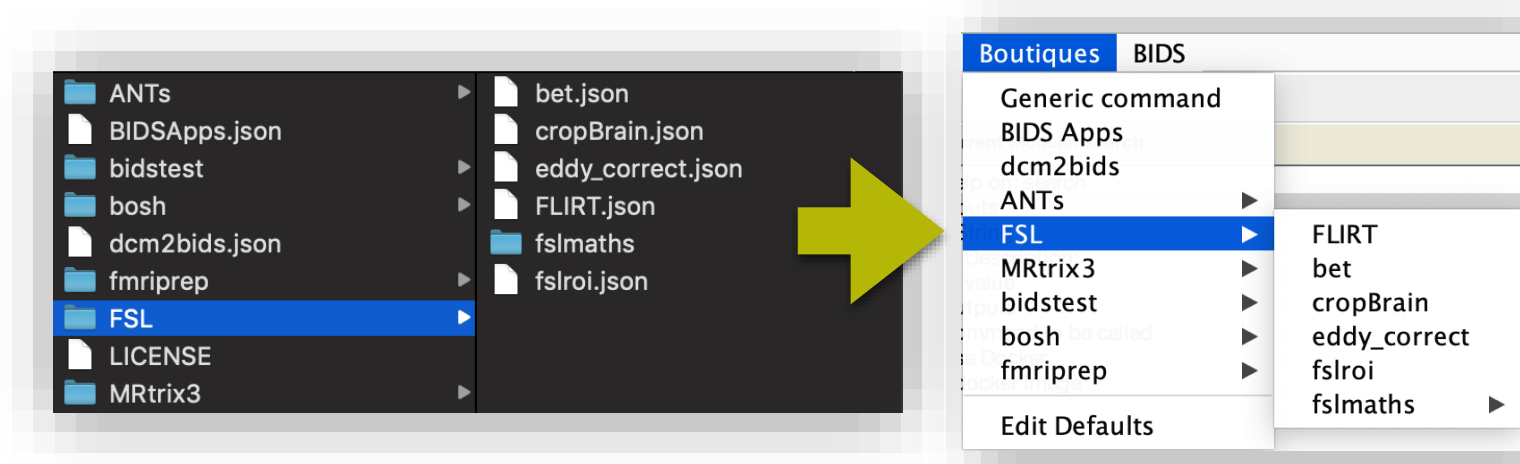


Boutiques app

Boutiques is the standard for describing the usage of any neuroimaging tools (e.g. FSL, ANTs) <https://boutiques.github.io/>

These tools are run in Docker: a software container technology (no more installation or dependencies issues)

► This figure shows the list of Boutiques descriptors that are already included in the package. drag and drop new descriptors in the « Boutiques » folder to have them available in SPM. New descriptors written by the community can be found using the command Bosh>Search and Bosh>Pull



Quality Control app

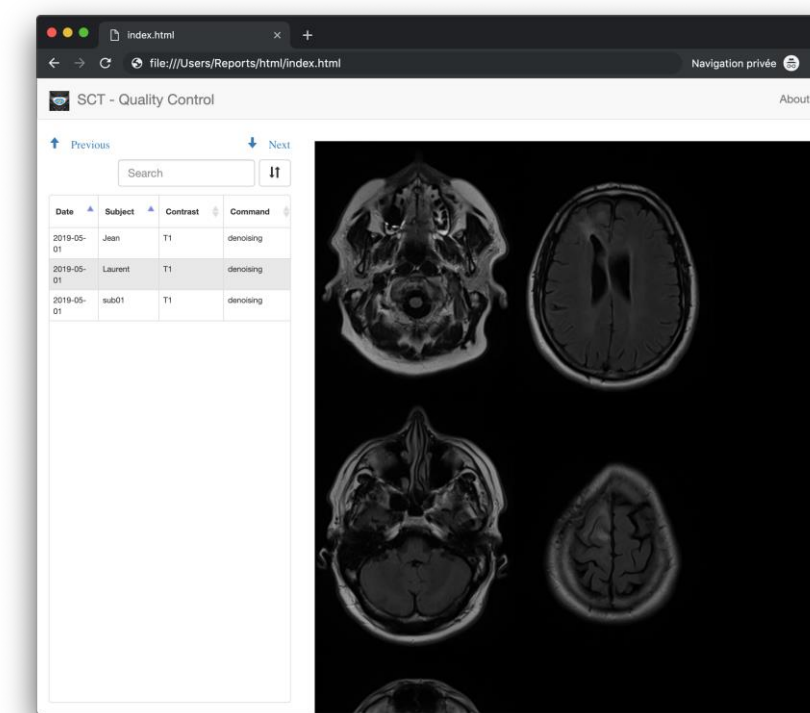
includes a module for exporting Nifti volumes as an interactive HTML page and an advanced Nifti viewer with ROI tools.

► **Export NIFTI into html reports**

Based on the template proposed by the active spinal cord toolbox team [6]

Instantly scroll though MRI sessions or subjects

Check segmentations or registrations



▼ **Advanced viewer with ROI tools**

- NIFTI volume (3D to 5D)
- Get image statistics
- Measurement tools (distances, volumes)
- Advanced, yet simple, tools to draw a mask (slice interpolation, active contour, smoothing)

