

# NICePy: A Web-based pipeline manager for processing neuroimaging data based on Nipype.

Dirk K. Müller<sup>1</sup>, René Küttner<sup>1</sup>, Ralf Hannig<sup>1</sup>, Thomas Frank<sup>1</sup>, Juliane Müller<sup>1</sup>, and Michael Marxen<sup>1</sup>

<sup>1</sup>Department of Psychiatry and Neuroimaging Center, Technische Universität Dresden, Dresden, 01187, Germany

## Introduction

NICePy stands for Neuroimaging Center Python Pipelines, a web-based pipeline manager developed at the Neuroimaging Center, Technische Universität Dresden, Germany. This software has been designed to offer standardized and parallelized imaging processing pipelines developed by experts to a community of applied scientists with an efficient and intuitive web-based interface. Within Neuroimaging, there is a vast amount of software, which enables use of algorithms from different software packages (e.g., Nipype [1], FSL [2], FreeSurfer [3], SPM [4] ...). Therefore, scientists must handle many tools, to get the most out of their data. To unify access to neuroimaging software, Nipype was developed. Nipype is an open source, flexible, lightweight, and extendible neuroimaging data processing framework in Python. It offers interfaces to many algorithms used in neuroimaging enabling scalability and parallelization. However, programming processing pipelines and interfaces in Nipype requires more advanced programming skills. We developed NICePy, a web-based software framework, to share selected processing pipelines with applied scientist and enable convenient access to such pipelines and acquired data.

## Methods and software design

The software framework comprises (Fig. 1):

- Sorting of neuroimaging data:** Sorting and setting of permissions is done in a hybrid approach semi-automated. For automated sorting, DICOM header information is used and additionally filtered by regular expressions handling incomplete or erroneous DICOM tags. Other data containing only sparse header information (e.g., physiological data) is sorted computer-assisted by programs that require users to provide project and subject numbers.
- An abstract pipeline class:** All pipeline steps are derived from an abstract pipeline defining the interface. Pipeline steps have a well-defined input and output interface, for imaging data, sequence parameters, processing parameters, onsets, etc. Data is stored in a container based on Python's dictionary type and converted to data formats as required by Nipype nodes. Pipeline steps can be queried for mandatory class methods and inputs. Additionally, each pipeline step is able to communicate its required inputs (images and parameters) to a web interface. Pipeline steps are connected together building a processing pipeline, e.g. including converting, preprocessing, single subject statistics, and group statistics. Processing can be parallelized on the subject level.
- A web-interface:** The web interface allows users with a password access to their projects. Subjects are assigned to projects allowing a fine-grained handling of permissions. Users can be assigned to different roles (e.g., admin, scientist ...) rendering certain pipelines visible for this user. The web interface allows a bulk configuration of pipelines for many subjects (Fig. 2). This makes it very easy to configure a large number of subjects in a single step. Fine-grained adjustments are possible, if needed.
- A database server:** The database is the hub of the system. The sorting software prepares the meta-data for the database, the web interface enters job configurations into the database, the workers (see below) extract this information and execute the processing job. The database comprises only meta-data necessary for processing (e.g., study, location of imaging data ...), but no personal or imaging data.
- NICePy workers** start and monitor the jobs. Workers can run decentralized on different computers. Running workers send regularly a heartbeat to the database, other workers will clean up and claim the job, if a worker dies unexpectedly.
- Quality Assurance** is done on three different Levels: a) Code quality using Python's unittest, b) Pipeline results are hashed for different combinations of pipeline steps and stored for later comparison. A report is created indication success, failed and missing pipeline steps. c) A web page showing tools for assessing motion correction, coregistration and normalization.

Processing was done on a computer with 2 processors, Intel(R) Xeon(R) CPU E5-2640 with 6 cores each and 24 threads in total using hyper threading, 128 GB Ram, 4 x 4 Gb hard disks (Raid 0). SPM-based pipelines utilize SPM standalone and work without a Matlab license.

## Results

A picture of the current web-interface is shown in Fig. 2. Piloting of the framework is currently in progress and includes pipelines for fMRI preprocessing for four projects and a Freesurfer segmentation for another. Two projects are using multi-echo sequences. Preprocessing of 500 fMRT scans including slice time correction, realignment, coregistration, segmentation (SPM), normalization and smoothing takes about one day, 50 Freesurfer segmentations can be performed per day on our machine. A single subject takes less than an hour for preprocessing and about 12 h for a Freesurfer segmentation. All details about pipeline settings are available to the user: A short description for choosing the pipeline, a detailed description as needed for the method section of a publication and a script containing the executed command enabling the user to exactly reproduce the pipeline.

## Conclusions and Outlook

### NICePy

- utilizes the effort of a big developer community (e.g., Nipype, FSL, SPM, ...),
- offers easy access to standard pipelines,
- allows applied scientists to use new pipelines immediately
- can utilize big computer clusters supported by Nipype plugins.

## References

- 1) K Gorgolewski, et al. SS. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in Python. *Front. Neuroinform.* 5:13
- 2) M Jenkinson, et al. FSL. *NeuroImage*, 62:782-90, 2012
- 3) Freesurfer. <http://surfer.nmr.mgh.harvard.edu/>
- 4) SPM. <http://www.fil.ion.ucl.ac.uk/spm/>

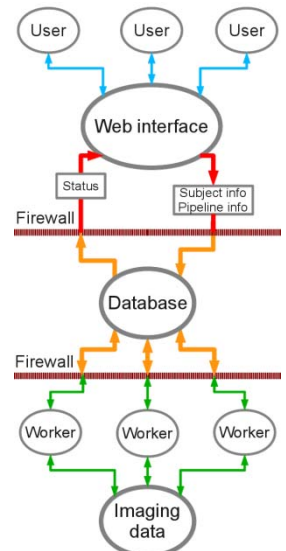


Fig. 1: Design concept of the NICePy.

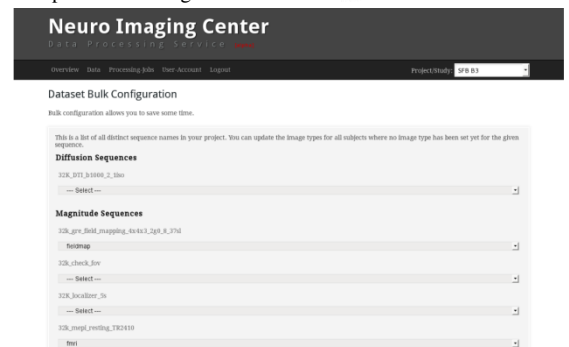


Fig. 2: Image of the bulk configuration tab. Image types have to be assigned to acquired scans according to pipeline requirements.