0136

Studying resting state connectivity using Nipype

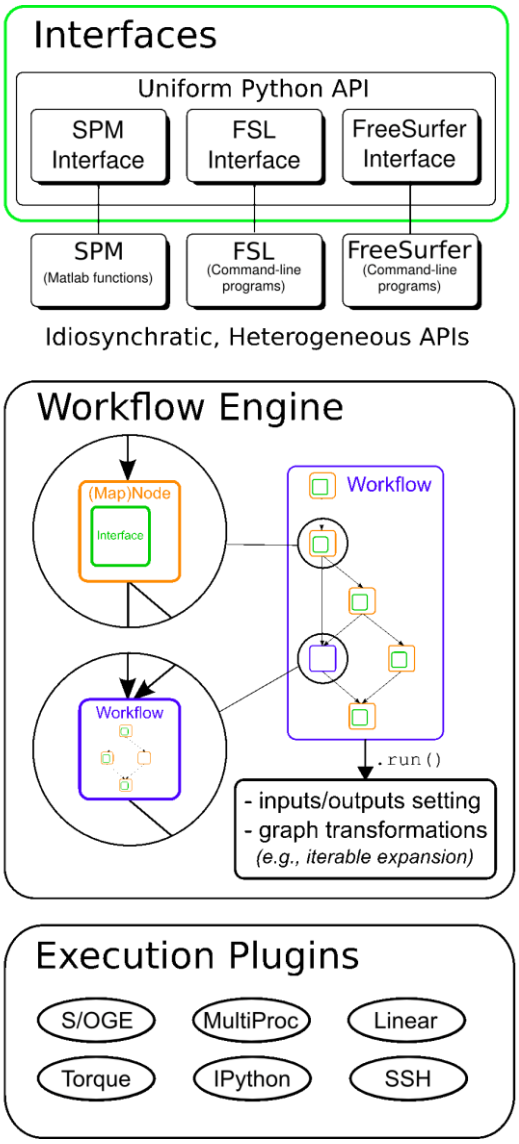
K. Gorgolewski[1], Y. Halchenko[2], M. Hanke[3], M. Notter[4], G. Varoquaux[5], M.L. Waskom[6], E. Ziegler[7], S. Ghosh[8]
[1]Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany
[2]Darthmoth College, Hanover, United States
[3]University of Magdeburg, Magdeburg, Germany
[4]University of Zurich, Zurich, Switzerland
[5]INRIA, Gif-sur-Yvette, France
[6]Stanford University, Stanford, United States
[7]Université de Liège, Liège, Belgium
[8]Massachusetts Institute of Technology, Cambridge, United States

Nipype is data processing framework written in Python. Its main goal is to provide a uniform access to many already existing neuroimaging tools. These currently include but are not limited to: FSL, SPM, FreeSurfer, and AFNI. At its core Nipype is a library of wrappers that take care of parsing and validating inputs, executing the software (whether it is a command line program, MATLAB or Python script), and collecting outputs (see Figure 1). The wrappers (also called Interfaces) can be used separately in an interactive fashion or can be combined into workflows. Nipype enables rapid prototyping of simple to complex data processing workflows. Workflows created in Nipype can be efficiently executed on a local machine or a computer cluster. The framework takes advantage of the information about the dependencies between the Interfaces to parallelize the data processing. Additionally nipype supports seamless parameter space exploration on any level of the data processing workflow.

Nipype excels by providing access to multiple preprocessing algorithms. For resting state data, one can choose slice timing, motion correction, smoothing, coregistration and temporal filtering algorithms from AFNI, FSL, FreeSurfer and SPM packages. Additionally, algorithms are available for artifact detection, CompCor and simultaneous slice timing and motion correction algorithm [1]. Nipype also supports ICA methods through MELODIC (FSL). Preprocessed resting state data can be easily mapped to regions and surfaces through FreeSurfer interfaces. Such resting state connectivity analyses can be constrained, visualised, and/or supplemented by anatomical connectivity estimated from Diffusion Weighted Imaging (DWI). Nipype supports DWI data processing through FSL, Camino, MRtrix, and Diffusion Toolkit. Several diffusion modelling and tractography algorithms are available. Both resting and diffusion data can be represented as graphs and analyzed using the Connectome Mapper Toolkit and NetworkX library. Nipype includes examples of applying predefined resting state workflows to freely available data (INDI, NKI-RS). Nipype provides a framework for reproducible analysis and allows efficient exploration of algorithms and their parameters in the context of connectivity and other brain imaging analysis.

[1] Roche, A. A four-dimensional registration algorithm with application to joint correction of motion and slice timing in fMRI. IEEE Trans Med Imaging. 2011 Aug;30(8):1546-54

figure 1


Interfaces
Uniform Python API
SPM Interface
FSL Interface
FreeSurfer Interface
SPM
(Matlab functions)
FSL
(Command-line programs)
FreeSurfer
(Command-line programs)
Idiosynchratic, Heterogeneous APIs
Workflow Engine
(Map)Node
Interface
Workflow
Workflow
.run()
- inputs/outputs setting
- graph transformations
(e.g., iterable expansion)
Execution Plugins
S/OGE
MultiProc
Linear
Torque
IPython
SSH