



## From Nipype to Pydra: a Clinica story

Ghislain Vaillant, Nicolas Gensollen, Matthieu Joulot, Omar El-Rifai,  
Mauricio Diaz, Olivier Colliot, Ninon Burgos

### ► To cite this version:

Ghislain Vaillant, Nicolas Gensollen, Matthieu Joulot, Omar El-Rifai, Mauricio Diaz, et al.. From Nipype to Pydra: a Clinica story. OHBM 2023 - Annual meeting of the Organization for Human Brain Mapping, Jul 2023, Montreal, Canada. hal-04278898

**HAL Id: hal-04278898**

**<https://inria.hal.science/hal-04278898>**

Submitted on 10 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# From Nipype to Pydra: a Clinica story

Ghislain Vaillant<sup>1</sup>, Nicolas Gensollen<sup>1</sup>, Matthieu Joulot<sup>1</sup>, Omar El Rifai<sup>1</sup>, Mauricio Diaz Melo<sup>1</sup>,  
Olivier Colliot<sup>1</sup>, Ninon Burgos<sup>1</sup>

<sup>1</sup> Sorbonne Université, Institut du Cerveau - Paris Brain Institute - ICM, CNRS, Inria, Inserm, AP-HP, Hôpital de la Pitié Salpêtrière, F-75013, Paris, France

## Introduction

Carrying out neuroimaging studies often involves multiple steps requiring different tools. As such it can be difficult to reproduce an experiment. Clinica (Routier et al., 2021) is a software platform that aims at empowering users with an environment to make reproducibility possible.

In this abstract, we describe Clinica's migration to Pydra, a modern dataflow engine developed for the Nipype project (<https://nipype.github.io/pydra>). We also highlight our wider contributions to the Pydra ecosystem which will benefit the community at large.

## Methods

Clinica provides processing pipelines which consume a BIDS dataset (Gorgolewski et al., 2016) as input and produce BIDS-like derivatives as output (Figure 1). Each pipeline may aggregate multiple workflows, whose complexity varies depending on the nature of the task, processed modalities and parameterization strategies.

Nipype (Gorgolewski et al., 2011) was not designed for easy composition of workflows to a larger pipeline. Workarounds exist but are limited to linear sub-graphs and their poor ergonomics increases accidental code complexity. Pydra (Jarecka et al., 2020) solves this issue by considering workflows a first-class construct for processing graphs, enabling arbitrary composition, splitting and combining of sub-graphs computation.

Another radical departure from Nipype is Pydra's support for third-party extensions. Pydra itself provides the task specifications and core execution engine only, whilst leaving a dedicated package namespace for tasks implementation.

Clinica's migration has been planned as follows:

- Experiment with Pydra's feature set by implementing a simple Clinica pipeline in a standalone project.
- Identify common pipeline construction and manipulation patterns by porting other pipelines to Pydra outside Clinica.
- Implement a first iteration of the Pydra-based engine by refactoring these patterns inside Clinica.
- Port the rest of the Clinica pipelines to the new engine, refactoring systematically as code changes. Pydra-based pipelines are made available in parallel to the legacy ones to test for regressions.
- Once porting is complete and code stabilizes, deprecate pipelines using the legacy engine in favor of those running with Pydra.

## Results

Initial efforts at porting our existing workflows to Pydra were facilitated by the ``pydra-nipype1`` plugin, which allows on-the-fly translation of individual Nipype nodes to Pydra tasks. Using this plugin allowed us to focus on factoring tasks into functional sub-workflows which could be tested and validated in isolation before integration.

For native Python tasks or tools missing a corresponding Nipype interface, the Pydra library was used to build our own task packages. Some of these task packages have been shared to the broader community, such as ``pydra-freesurfer`` for FreeSurfer (Fisch et al., 2012) and ``pydra-bids`` for BIDS reading and writing.

A number of bugs have been found in the Pydra code base during our migration process. In addition to providing fixes for these bugs, we have been actively engaging in the development of the project. Our engagement should benefit both Clinica and the Pydra community at large.

Most pipelines have now been migrated to the new Pydra-based engine. Each pipeline follows a simple architecture composed of a sequence of input, core and output workflows, whose execution is parameterized through the command-line (Figure 2). Given our progress and help received from the community so far, we are hopeful to complete the migration within the next few months.

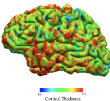
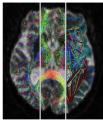
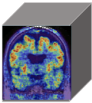
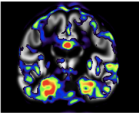
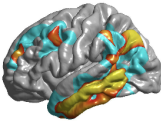

## Conclusions

We went on a journey to upgrade Clinica's engine to Pydra and benefit from its enhanced workflow definition capabilities and extensible design through plugins. Although migration is still on-going, initial results are very promising and lead to significant design simplification and improved code re-use. Once complete, we will explore Pydra's capabilities to run workflows in different environments, such as the Cloud.

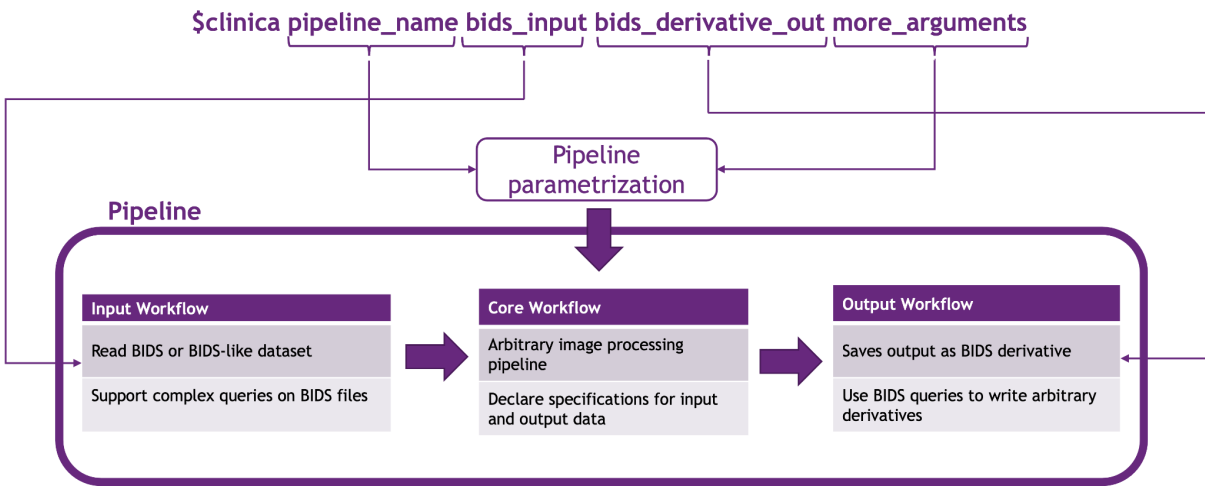
## References

- Fischl, B. (2012). 'FreeSurfer'. *Neuroimage*, 62(2), 774-781.
- Gorgolewski, K. J. et al. (2016). 'The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments'. *Scientific data*, 3(1), 1-9.
- Gorgolewski, K. J. et al. (2011). 'Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python'. *Frontiers in Neuroinformatics*. 2011;5.
- Jarecka, D. et al. (2020). "Pydra - a flexible and lightweight dataflow engine for scientific analyses." *Proceedings of the 19th python in science conference*. Vol. 132. 2020.
- Routier, A. et al. (2021). 'Clinica: an open-source software platform for reproducible clinical neuroscience studies'. *Frontiers in Neuroinformatics*, 15.

Figures

<div>Anatomical MRI</div> <div><ul style="list-style-type: none"><li><code>t1 flair-linear</code>: Bias field correction, affine registration and cropping for T1 and FLAIR</li><li><code>t1-volume</code>: Tissue segmentation (GM, WM, CSF), inter-subject registration using Dartel, spatial normalization to standard space (MNI)</li><li><code>t1-freesurfer/t1-freesurfer-longitudinal</code>: Cortical surface extraction, segmentation of subcortical structures, cortical thickness estimation, spatial normalization to standard space (FsAverage)</li></ul></div> <div></div>	<div>Diffusion MRI</div> <div><ul style="list-style-type: none"><li><code>dwi-preprocessing</code>: Correction of raw DWI data</li><li><code>dwi-dti</code>: Extraction of DTI-based measures, normalization to standard space (MNI)</li><li><code>dwi-connectome</code>: Tractography &amp; connectome</li></ul></div> <div></div>	<div>PET</div> <div><ul style="list-style-type: none"><li><code>pet-linear</code>: Affine registration, intensity normalization and cropping</li><li><code>pet-volume</code>: Registration to T1 MRI, partial volume correction, spatial normalization to standard space (MNI), intensity normalization</li><li><code>pet-surface/pet-surface-longitudinal</code>: Registration to T1MRI, Intensity normalization, partial volume correction, projection to cortical surface, spatial normalization to standard space</li></ul></div> <div></div>
<div>Machine Learning</div> <div><ul style="list-style-type: none"><li><code>machinelearning-prepare-spatial-svm</code>: Preparation of T1 MRI and PET data for spatially regularized SVM</li><li><code>machinelearning-classification</code>: Classification based on machine learning using scikit-learn</li></ul></div> <div></div>	<div>Statistics</div> <div><ul style="list-style-type: none"><li><code>statistics-volume</code>: voxel-based mass univariate analysis with SPM</li><li><code>statistics-surface</code>: Surface-based mass-univariate analysis with SurfStat</li></ul></div> <div></div>	<div>Converters</div> <div><p>Several pipelines to convert public neuroimaging datasets into the BIDS standard</p><ul style="list-style-type: none"><li><code>adni-to-bids</code></li><li><code>nifd-to-bids</code></li><li><code>alibi-to-bids</code></li><li><code>habs-to-bids</code></li><li><code>oasis-to-bids</code></li><li><code>oasis3-to-bids</code></li><li><code>ukb-to-bids</code></li><li><code>genfi-to-bids</code></li></ul></div> <div></div>

**Figure 1.** List of the main functionalities currently available in Clinica: converters of public neuroimaging datasets to BIDS; neuroimaging processing pipelines for anatomical MRI, diffusion MRI and PET; and statistics and machine learning workflows.



**Figure 2.** A typical Clinica pipeline under Pydra. Each pipeline follows a simple architecture composed of a sequence of input, core and output workflows, whose execution is parameterized through the command-line.