



# Pypes: Workflows for Processing Multimodal Neuroimaging Data

Alexandre M. Savio<sup>1,2\*</sup>, Michael Schutte<sup>1</sup>, Manuel Graña<sup>2,3</sup> and Igor Yakushev<sup>1,4</sup>

<sup>1</sup> Department of Nuclear Medicine, Klinikum Rechts der Isar, Technische Universität München, Munich, Germany, <sup>2</sup> Asociación Python San Sebastián (ACPySS), San Sebastián, Spain, <sup>3</sup> Computational Intelligence Group, University of the Basque Country (UPV/EHU), San Sebastián, Spain, <sup>4</sup> Neuroimaging Center at Technische Universität München (TUM-NIC), Technische Universität München, Munich, Germany

**Keywords:** python, brain imaging, PET, MRI, registration, denoising, image analysis, brain connectivity

## 1. INTRODUCTION

Every year, enormous amounts of scientific data are made available to the public (Poline et al., 2012). This trend is due to an increasing demand for transparency, efficiency, and reproducibility. Neuroimaging is a salient example of this trend.

In response to the growing concern about the need of publishing relevant software codes (Ince et al., 2012) in the context of results' reproducibility, there is an increasing number of open source initiatives that support code distribution and co-development (Halchenko and Hanke, 2012). The growing diversity of imaging modalities demand from the practitioner a deep technical knowledge of data pre- and post-processing. Consequently, there are open and free tools facilitating image data analysis, e.g., the Python module Nipype<sup>1</sup>. It offers a homogeneous programming interface and integrates many of these data processing tools. In this sense, resting-state functional magnetic resonance imaging (rsfMRI) is receiving considerable attention by the community with tools such as the Configurable Pipeline for the Analysis of Connectomes (C-PAC)<sup>2</sup>, and the Data Processing Assistant for Resting-State fMRI (DPARSF)<sup>3</sup>.

As a further contribution to this development, this paper presents a new Python module Pypes—<https://github.com/Neurita/pypes>. It includes a collection of workflows, reusable neuroimaging pipelines using Nipype, along with some utilities. This library seeks to simplify the reusability and reproducibility of multimodal neuroimaging studies, offering pre- and post-processing utilities inspired by C-PAC. It pre-processes Positron Emission Tomography (PET) and three MRI-based modalities: structural, rsfMRI, and diffusion-tensor MRI (DTI). It also shares an easy-to-use pipeline for COBRE<sup>4</sup>, a public available dataset. Pypes has been motivated by a need for efficient and reproducible brain PET/MRI data processing methods. Namely, hybrid PET/MRI scanners become a relevant source of multimodal imaging data, posing new computational challenges. For instance, a simultaneous measurement of brain glucose metabolism and functional connectivity (Aiello et al., 2015; Riedl et al., 2016) opens new perspectives in neuroscience. Structural, functional, and metabolic imaging protocols have been proposed for clinical evaluation of dementia and neuro-oncological cases (Werner et al., 2015; Henriksen et al., 2016). Pypes' immediate motivation was to process PET/MRI data from an ongoing study with more than 400 subjects with suspected neurodegenerative disorders.

## OPEN ACCESS

### Edited by:

Juan Manuel Gorriz,  
University of Granada, Spain

### Reviewed by:

Bogdan Raducanu,  
Autonomous University of Barcelona,  
Spain

Michał Wozniak,  
Wrocław University of Technology,  
Poland

### \*Correspondence:

Alexandre M. Savio  
[alex.savio@tum.de](mailto:alex.savio@tum.de)

**Received:** 21 February 2017

**Accepted:** 17 March 2017

**Published:** 11 April 2017

### Citation:

Savio AM, Schutte M, Graña M and Yakushev I (2017) Pypes: Workflows for Processing Multimodal Neuroimaging Data. *Front. Neuroinform.* 11:25. doi: 10.3389/fninf.2017.00025

<sup>1</sup><http://nipype.readthedocs.io/>

<sup>2</sup><http://fcp-indi.github.io/>

<sup>3</sup><http://www.rfmri.org/DPARSF>

<sup>4</sup>[http://fcon\\_1000.projects.nitrc.org/indi/retro/cobre.html](http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html)

The paper is organized as follows. After introducing the Python neuroimaging ecosystem and specifically Nipype, we show how to prepare image data for the workflows available in Pypes. Then, we describe workflow configuration for specific imaging modalities. Finally, we present the Pypes pre-processing pipelines and the post-processing utilities. We finish the paper with conclusions and future developments.

## 2. THE SOFTWARE ECOSYSTEM

The neuroimaging open software ecosystem was born more than 10 years ago, with brain image processing tools such as Statistical Parametric Mapping (SPM12) (Ashburner, 2012), the FMRIB Software Library (FSL) (Jenkinson et al., 2012), AFNI (Analysis of Functional NeuroImages) (Cox, 2012), and FreeSurfer (Fischl, 2012). These libraries were developed by universities and government institutions, and published under open-source or free-software licenses. They have boosted the neuroimaging research, acting as a seed to a now flourishing software ecosystem with new actors appearing lately, e.g., Advanced Normalization Tools (ANTs) (Avants et al., 2014), PET partial volume correction (PETPVC) (Thomas et al., 2016), and MRtrix (Tournier et al., 2012).

As a programming language, Python is becoming very popular. Its supporting community of users and developers is making a great effort to spread good practices of the development process. As a result, there is a booming variety of libraries, guidelines, documentation, development tools, software testing, and continuous integration. A broad collection of tools is available, ranging from general numerical libraries<sup>5</sup>, to specific applications such as machine learning in Scikit-Learn (Pedregosa et al., 2011), or deep neural networks in Tensorflow<sup>6</sup>.

In the neuroscientific field, a group of projects have joined in a community of practice called Nipy<sup>7</sup>. From Nipy, it's possible to find projects to access different neuroimaging file formats<sup>8</sup>, diffusion brain MRI with Dipy (Garyfallidis et al., 2014), statistical learning and fMRI with Nilearn (Abraham et al., 2014) as well as building processing pipelines with Nipype, and others (Gramfort et al., 2013).

Nipype allows to interact and combine tools from different software packages, some already mentioned before, facilitating faster data processing by running the workflows in parallel on many cores/machines. Nipype makes your analyses easily reproducible allowing to share your processing workflows with the community, it also captures the pipelines provenance information in a formal and rigorous way, and allows to restart the pipelines if something has gone wrong. It already presents examples of processing workflows for many imaging modalities. However, it is still a challenge for a non-expert programmer to prepare the data, programme the data selection, configure the workflow, and run it. Built on top of Nipype, Pypes provides workflows for multimodal brain MRI and PET/MRI. We verified

the correct working of these pipelines over our clinical dataset of more than 400 subjects and the COBRE database. Pypes uses Nipype as interface to all the command line and Matlab neuroimaging tools. It presents a software structure with a clear way to concatenate and attach new workflows, offering a simple way to express the input data folder structure, and easily configure each node. The output of the workflows are structured in the same way as the input folder structure, and the output file name conventions are homogeneous and clear. The users would have to structure their data following recommendations and use the provided examples of Python functions to build their own pipeline.

## 3. DATA PREPARATION AND PIPELINE CONFIGURATION

Pypes expects neuroimage files to be in the NifTI format. Currently, a good tool to convert from DICOM to NifTI is `dcm2niix`<sup>9</sup>.

One practical feature that Pypes adds to Nipype is the management of file input and output. The file input selection system is clearer and more flexible than that of NiPype using the parametric file paths provided by the `hansel` package<sup>10</sup>. The output will have the same folder structure as the input and the naming convention is uniform across the pipelines.

The main nodes in the pipelines are configurable through a configuration file. We recommend using the YAML (<https://en.wikipedia.org/wiki/YAML>) format for this file. To change the default value of a node parameter one has to add to the configuration file an entry for the value you want. For example, let's say we have a `spm.Normalize12` node named `anat_warp` in one of the workflows. We want to set the value of the parameter `bias_regularization` to `0.1`. We have to add an entry to the configuration file as:

```
anat_warp.bias_regularization: 0.1
```

Pypes includes in its source code an example of configuration file with the main node settings and explanations.

## 4. PRE-PROCESSING METHODS

Each image modality has certain kinds of artifacts and noise, such that specific correction procedures are needed. Multimodal information requires data fusion. Here, we explain the main features of each pre-processing pipeline.

### 4.1. Anatomical MRI

First, we needed to process structural T1-weighted MRI images. We prepared a pipeline that first performs a bias-field correction using ANTs' `N4BiasFieldCorrection` (Tustison et al., 2010). Then it segments the brain tissues and registers the image to a standard template space (MNI) with the SPM12 New Segment tool. Finally, it creates a brain mask based on the

<sup>5</sup><http://www.scipy.org/>

<sup>6</sup><https://www.tensorflow.org>

<sup>7</sup><http://nipy.org/>

<sup>8</sup><http://nipy.org/nibabel>.

<sup>9</sup><https://github.com/rordenlab/dcm2niix>

<sup>10</sup><https://github.com/alexsavio/hansel>.

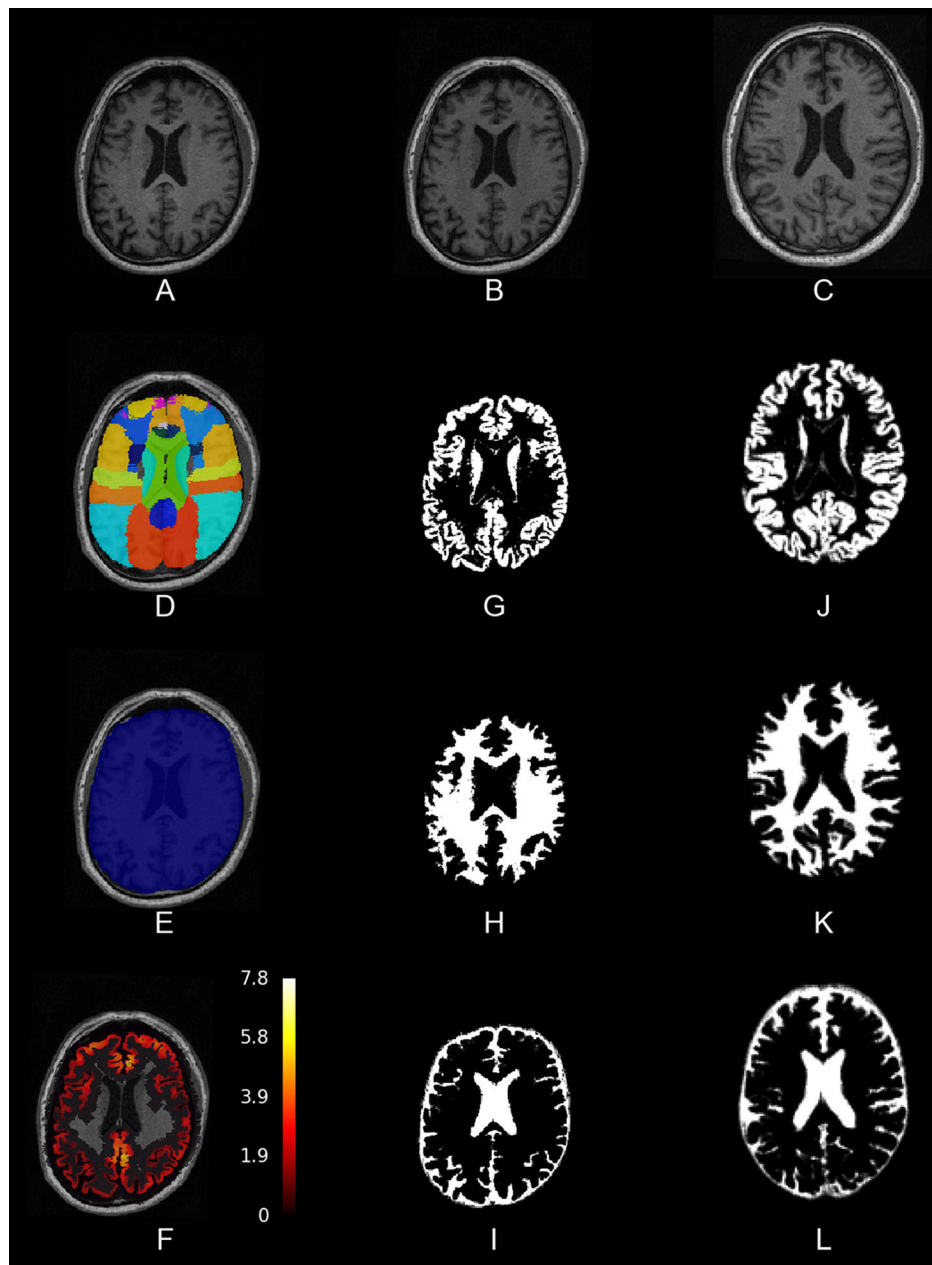
tissue segmentations. In addition, we needed to warp an atlas, or any other image in the SPM12 standard space, to the subject's native space for further analyses. In **Figure 1**, we present an image output of this pipeline. It is implemented in `pypes.anat.preproc.spm_anat_preprocessing`.

## 4.2. Positron Emission Tomography

The computational analysis in the reference clinical study requires pre-processing of FDG-PET images that had been

acquired simultaneously with MRI data (Savio et al., 2017). Pypes offers two main pipelines for PET data, one with and the other without involving T1-weighted MRI. Both pipelines use SPM12 Normalize to warp PET images to the MNI space. The latter option requires a group-template pipeline, where a group template is created from all subjects, and then all PET images are normalized to this group template.

The PET/MR pipeline also applies PVC and normalizes structural MRI data to the PET space. It is implemented



**FIGURE 1 | Slices of one example of the anatomical pipeline on one sample image. (A)** The raw MPRAGE image, **(B)** the bias-field corrected MPRAGE, **(C)** the bias-field corrected MPRAGE in MNI space, **(D)** the Hammers atlas in anatomical space, **(E)** a brain mask, and **(F)** the result from the cortical thickness pipeline, **(G–I)** gray matter, white matter and cerebro-spinal fluid (tissue segmentations), and **(J–L)** tissue segmentations in MNI space.

in `pypes.pet.mrpet.spm_mrpet_preprocessing`. In **Figure 2**, we present an image output of this pipeline.

#### 4.2.1. Partial Volume Correction (PVC)

One of the main issues with PET is signal degradation caused by partial volume effects (PVE). PVC methods improve the quantitative accuracy of PET data by compensating for the noise owing to a limited scanner resolution (Erlandsson et al., 2012), specially in the atrophied brains. Generally, brain anatomical information from MRI is used here. The recently published PETPVC (Thomas et al., 2016) library<sup>11</sup> provides eight core methods of PVC in an open source tool.

### 4.3. Functional MRI

The referred clinical study investigates functional connectivity, so we implemented a configurable rsfMRI data processing pipeline. For this, we need anatomical localization for nuisance corrections based on tissue signal, so this pipeline is connected to the MPRAGE processing pipeline (Section 4.1). The first part of this pipeline, implemented in `pypes.fmri.clean.fmri_cleanup_wf`, trims the first 6 s from the data, carries out slice-time correction based on SPM12 SliceTiming, correction for motion with Nipy's SpaceTimeRealigner (Roche, 2011), co-registration of the tissues from anatomical space to fMRI space, correction for nuisances extracted from time-course SNR (TSNR) estimation, artifact detection from Nipype's rapidART, motion correction, signal

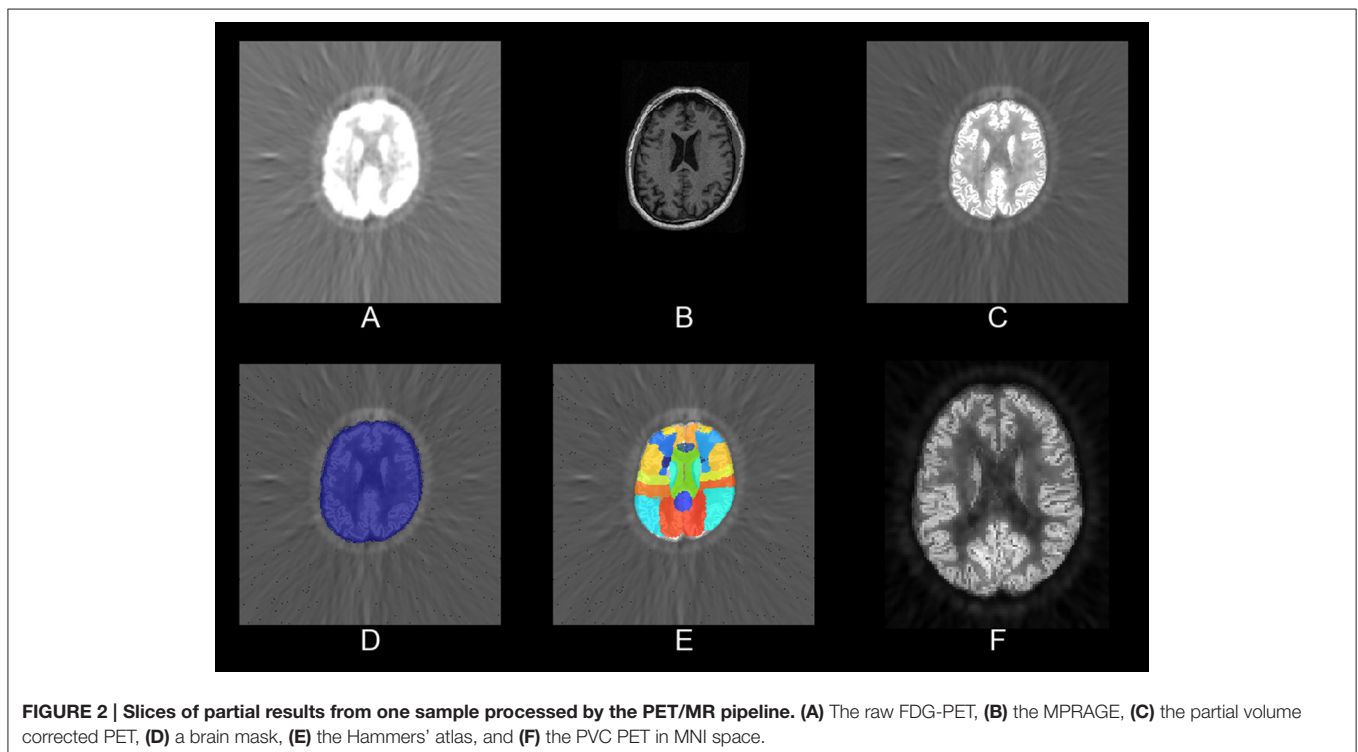
component regression from different tissues (CSF, WM, and/or GM), and optionally, global signal regression. The trends detected from these filters are regressed out from the fMRI data. Each of these corrections are optional and configurable. After the nuisance correction step, a bandpass time filter is applied to extract resting-state frequencies and the data is spatially smoothed. In the second step of this pipeline, implemented in `pypes.fmri.warp.spm_warp_fmri_wf`, the main outputs of the first are warped to MNI using SPM12 Normalize.

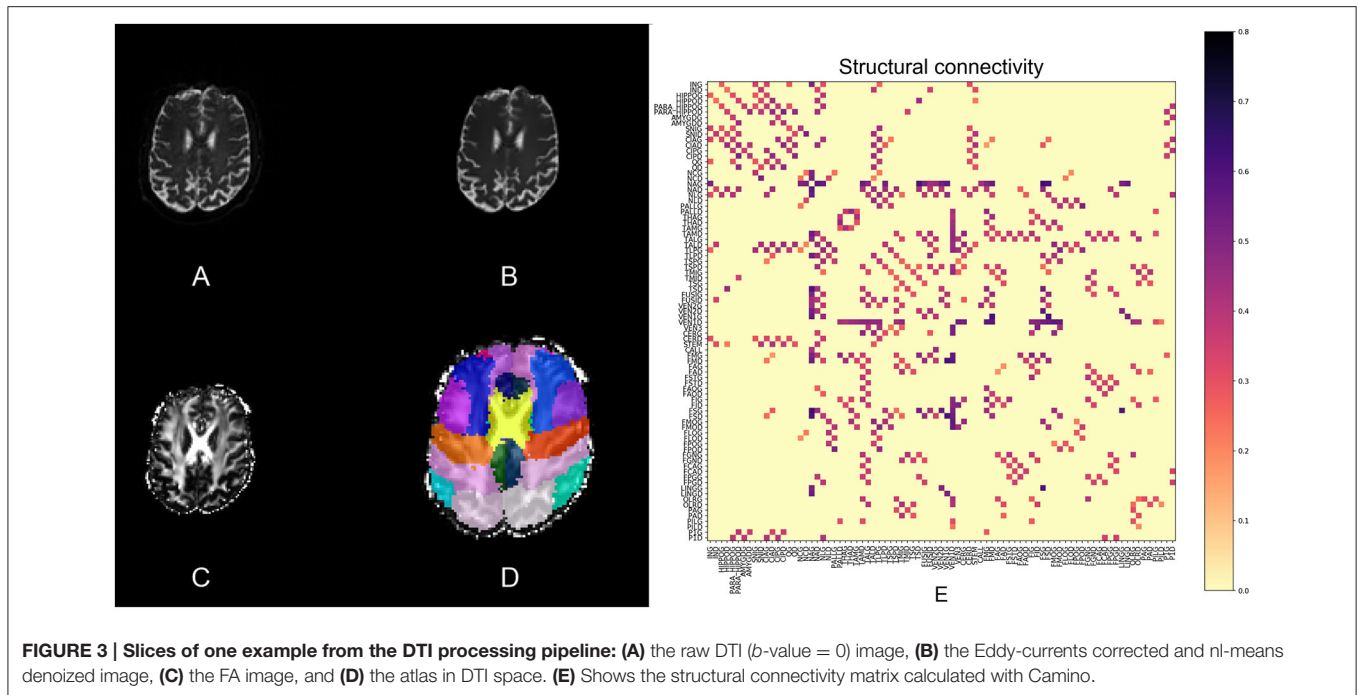
### 4.4. Diffusion-Tensor MRI (DTI)

DTI may be useful to support or extend findings of metabolic and functional connectivity. We provide a pipeline that performs DTI correction and pre-processing, tensor-fitting, and tractography. This pipeline is implemented in `pypes.dmri.dti.attach_spm_fsl_dti_preprocessing`. First it uses FSL Eddy (Andersson and Sotiropoulos, 2016) for Eddy currents and motion correction. Then Non-local Means (Coupe et al., 2008) is used from Dipy for image de-noising with a Rician filter. This pipeline also estimates motion statistics with Nipype's RapidArt for *post-hoc* quality check, co-registers the anatomical image to diffusion space, and rotates the b-vectors based on motion estimation from Eddy. Optionally, it will warp an atlas to diffusion space (for further tractography).

An extra pipeline implemented in `pypes.dmri.camino.camino_tractography` uses Camino (Friman et al., 2006) to calculate Fractional Anisotropy (FA) and perform ROI-to-ROI deterministic tractography using Track (Basser et al.,

<sup>11</sup><https://github.com/UCL/PETPVC>.





**FIGURE 3 | Slices of one example from the DTI processing pipeline: (A)** the raw DTI ( $b$ -value = 0) image, **(B)** the Eddy-currents corrected and nl-means denoised image, **(C)** the FA image, and **(D)** the atlas in DTI space. **(E)** Shows the structural connectivity matrix calculated with Camino.

1994). The tractography pipeline exports two connectivity matrices: one with the number of tracts and the other with average tract FA values, for each pair of ROIs. **Figure 3** illustrates the DTI pipeline, including the connectivity matrix.

## 5. POST-PROCESSING METHODS

Pypes offers extra utilities and shorter *post-hoc* pipelines. It provides an Independent Component Analysis (ICA) interface to use Nilearn's CanICA and DictLearning against individual or group of fMRI data. These pipelines are implemented in `pypes.postproc.decompose`. For the analysis of ICA results and resting state networks, we implemented different similarity measures in `pypes.ica.spatial_maps`, including Goodness of fit (Zhou et al., 2010), Pearson's correlation, and many others. A cortical thickness pipeline based on ANTs (Tustison et al., 2014) is implemented in `pypes.anat.cortex`.

We created plotting utilities to visualize and publish the pre- and post-processing results. In `pypes.ica.plotting`, we implemented a series of methods to plot ICA spatial maps and loading coefficients from Nilearn and GIFT<sup>12</sup>.

## 6. CONCLUSION AND FUTURE WORK

We have published Pypes, a tool that allows an easy configuration of workflows for neuroimage analysis using

resources from state of the art open source libraries, while including new handy facilities for input/output data configuration and plotting. Pypes gives support to the growing community of researchers having access to hybrid PET/MRI scanners which require complex preprocessing and analysis of multimodal imaging data. For more details on how to use Pypes, check the documentation on <http://neuro-pypes.readthedocs.io/>.

Some aspects of PET/MRI processing are still to be considered in Pypes. A synchronized acquisition allows the use of MRI information for attenuation correction (AC) of PET images, though there is no consensus on the best algorithm (Cabello et al., 2016; Mehranian et al., 2016; Ladefoged et al., 2017). Future works will include the proposed AC algorithms in the PET/MRI pipelines of Pypes. Also other non-linear registration tools and DTI tractography methods should be added. Easy connection to machine learning libraries such as Nilearn and scikit-learn would allow further automatization of analyses and creation of predictive models for e.g., disease detection.

## AUTHOR CONTRIBUTIONS

AS, MS, and IY did the software requirements, design; MS and AS did the programming; MG contributed in the writing process. All authors contributed in drafting the work and revising it critically for important intellectual content; all authors give final approval of the version to be published; all authors agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

<sup>12</sup><http://mialab.mrn.org/software/gift/>.

## REFERENCES

- Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., et al. (2014). Machine learning for neuroimaging with scikit-learn. *Front. Neuroinform.* 8:14. doi: 10.3389/fninf.2014.0001400053
- Aiello, M., Salvatore, E., Cachia, A., Pappatà, S., Cavaliere, C., Prinster, A., et al. (2015). Relationship between simultaneously acquired resting-state regional cerebral glucose metabolism and functional MRI: a PET/MR hybrid scanner study. *NeuroImage* 113, 111–121. doi: 10.1016/j.neuroimage.2015.03.01700022
- Andersson, J. L. R., and Sotiropoulos, S. N. (2016). An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging. *NeuroImage* 125, 1063–1078. doi: 10.1016/j.neuroimage.2015.10.019
- Ashburner, J. (2012). SPM: a history. *NeuroImage* 62, 791–800. doi: 10.1016/j.neuroimage.2011.10.02500107
- Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., and Gee, J. C. (2014). The Insight ToolKit image registration framework. *Front. Neuroinform.* 8:44. doi: 10.3389/fninf.2014.00044
- Basser, P. J., Mattiello, J., and Lebihan, D. (1994). Estimation of the effective self-diffusion tensor from the NMR spin echo. *J. Magn. Reson. B* 103, 247–254. doi: 10.1006/jmrb.1994.1037
- Cabello, J., Lukas, M., Rota Kops, E., Ribeiro, A., Shah, N. J., Yakushev, I., et al. (2016). Comparison between MRI-based attenuation correction methods for brain PET in dementia patients. *Eur. J. Nucl. Med. Mol. Imaging* 43, 2190–2200. doi: 10.1007/s00259-016-3394-5
- Coupe, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., and Barillot, C. (2008). An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Trans. Med. Imaging* 27, 425–441. doi: 10.1109/TMI.2007.906087
- Cox, R. W. (2012). AFNI: what a long strange trip it's been. *NeuroImage* 62, 743–747. doi: 10.1016/j.neuroimage.2011.08.05600097
- Erlundsson, K., Buvat, I., Pretorius, P. H., Thomas, B. A., and Hutton, B. F. (2012). A review of partial volume correction techniques for emission tomography and their applications in neurology, cardiology and oncology. *Phys. Med. Biol.* 57:R119. doi: 10.1088/0031-9155/57/21/R119
- Fischl, B. (2012). FreeSurfer. *NeuroImage* 62, 774–781. doi: 10.1016/j.neuroimage.2012.01.021
- Friman, O., Farneback, G., and Westin, C. F. (2006). A Bayesian approach for stochastic white matter tractography. *IEEE Trans. Med. Imaging* 25, 965–978. doi: 10.1109/TMI.2006.877093
- Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Van Der Walt, S., Descoteaux, M., et al. (2014). Dipy, a library for the analysis of diffusion MRI data. *Front. Neuroinform.* 8:8. doi: 10.3389/fninf.2014.00008
- Gramfort, A., Luessi, M., Larson, E., Engemann, D., Strohmeier, D., Brodbeck, C., et al. (2013). MEG and EEG data analysis with MNE-python. *Front. Neurosci.* 7:267. doi: 10.3389/fnins.2013.00267
- Halchenko, Y. O., and Hanke, M. (2012). Open is not enough. Let us take the next step: an integrated, community-driven computing platform for neuroscience. *Front. Neuroinform.* 6:22. doi: 10.3389/fninf.2012.00022
- Henriksen, O. M., Marner, L., and Law, I. (2016). Clinical PET/MR imaging in dementia and neuro-oncology. *PET Clin.* 11, 441–452. doi: 10.1016/j.cpet.2016.05.003
- Ince, D. C., Hatton, L., and Graham-Cumming, J. (2012). The case for open computer programs. *Nature* 482, 485–488. doi: 10.1038/nature10836
- Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W., and Smith, S. M. (2012). FSL. *NeuroImage* 62, 782–790. doi: 10.1016/j.neuroimage.2011.09.015
- Ladefoged, C. N., Law, I., Anazodo, U., Lawrence, K. S., Izquierdo-Garcia, D., Catana, C., et al. (2017). A multi-centre evaluation of eleven clinically feasible brain pet/mri attenuation correction techniques using a large cohort of patients. *NeuroImage* 147, 346–359. doi: 10.1016/j.neuroimage.2016.12.010
- Mehranian, A., Arabi, H., and Zaidi, H. (2016). Quantitative analysis of MRI-guided attenuation correction techniques in time-of-flight brain PET/MRI. *NeuroImage* 130, 123–133. doi: 10.1016/j.neuroimage.2016.01.060
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Poline, J.-B., Breeze, J., Ghosh, S., Gorgolewski, K., Halchenko, Y., Hanke, M., et al. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.* 6:9. doi: 10.3389/fninf.2012.00009
- Riedl, V., Utz, L., Castrillón, G., Grimmer, T., Rauschecker, J. P., Ploner, M., et al. (2016). Metabolic connectivity mapping reveals effective connectivity in the resting human brain. *Proc. Natl. Acad. Sci. U.S.A.* 113, 428–433. doi: 10.1073/pnas.1513752113
- Roche, A. (2011). A four-dimensional registration algorithm with application to joint correction of motion and slice timing in fMRI. *IEEE Trans. Med. Imaging* 30, 1546–1554. doi: 10.1109/TMI.2011.2131152
- Savio, A., Fünfer, S., Tahmasian, M., Rachakonda, S., Manoliu, A., Sorg, C., et al. (2017). Resting state networks as simultaneously measured with fMRI and PET. *J. Nucl. Med.* doi: 10.2967/jnumed.116.185835. [Epub ahead of print].
- Thomas, B. A., Cuplov, V., Bousse, A., Mendes, A., Thielemans, K., Hutton, B. F., et al. (2016). PETPVC: a toolbox for performing partial volume correction techniques in positron emission tomography. *Phys. Med. Biol.* 61:7975. doi: 10.1088/0031-9155/61/22/7975
- Tournier, J.-D., Calamante, F., and Connelly, A. (2012). MRtrix: Diffusion tractography in crossing fiber regions. *Int. J. Imaging Syst. Technol.* 22, 53–66. doi: 10.1002/ima.22005
- Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., et al. (2010). N4itk: improved N3 bias correction. *IEEE Trans. Med. Imaging* 29, 1310–1320. doi: 10.1109/TMI.2010.2046908
- Tustison, N. J., Cook, P. A., Klein, A., Song, G., Das, S. R., Duda, J. T., et al. (2014). Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements. *NeuroImage* 99, 166–179. doi: 10.1016/j.neuroimage.2014.05.044
- Werner, P., Barthel, H., Drzezga, A., and Sabri, O. (2015). Current status and future role of brain PET/MRI in clinical and research settings. *Eur. J. Nucl. Med. Mol. Imaging* 42, 512–526.
- Zhou, J., Greicius, M. D., Gennatas, E. D., Growdon, M. E., Jang, J. Y., Rabinovici, G. D., et al. (2010). Divergent network connectivity changes in behavioural variant frontotemporal dementia and Alzheimer's disease. *Brain* 133, 1352–1367. doi: 10.1093/brain/awq075

**Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Savio, Schutte, Graña and Yakushev. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.