PyHRF: A Python Library for the Analysis of fMRI Data Based on Local Estimation of the Hemodynamic Response Function

Jaime Arias, Philippe Ciuciu, Michel Dojat, Florence Forbes, Aina Frau-Pascual, Thomas Perret, Jan M Warnking

HAL Id: hal-01566457
https://hal.science/hal-01566457

Submitted on 21 Jul 2017

PyHRF: A Python Library for the Analysis of fMRI Data Based on Local Estimation of the Hemodynamic Response Function

Jaime Arias‡†∗, Philippe Ciuciu§†, Michel Dojat¶‖†, Florence Forbes‡†, Aina Frau-Pascual‡†, Thomas Perret‡†, Jan M. Warnking¶‖†

✦

Abstract—Functional Magnetic Resonance Imaging (fMRI) is a neuroimaging technique that allows the non-invasive study of brain function. It is based on the hemodynamic variations induced by changes in cerebral synaptic activity following sensory or cognitive stimulation. The measured signal depends on the variation of blood oxygenation level (BOLD signal) which is related to brain activity: a decrease in deoxyhemoglobin concentration induces an increase in BOLD signal. The BOLD signal is delayed with respect to changes in synaptic activity, which can be modeled as a convolution with the Hemodynamic Response Function (HRF) whose exact form is unknown and fluctuates with various parameters such as age, brain region or physiological conditions.

In this paper we present PyHRF, a software to analyze fMRI data using a Joint Detection-Estimation (JDE) approach. It jointly detects cortical activation and estimates the HRF. In contrast to existing tools, PyHRF estimates the HRF instead of considering it as a given constant in the entire brain. Here, we present an overview of the package and showcase its performance with a real case in order to demonstrate that PyHRF is a suitable tool for clinical applications.

Index Terms—BOLD response, fMRI, hemodynamic response function

Introduction

Neuroimaging techniques, such as functional Magnetic Resonance Imaging (fMRI), allow the *in vivo* study of brain function by measuring the cerebral response to sensory or cognitive stimulation. For more than 20 years, the Blood-Oxygen-Level-Dependent (BOLD) fMRI modality has being the technique most used by neuroscientists to map the main functional regions and their links in the healthy and diseased brain.

The BOLD signal [OLKT90] reflects the changes in deoxyhemoglobin concentration in the brain. Briefly, when brain activity increases, local oxygen consumption in brain tissue increases, slightly increasing the concentration of deoxyhemoglobin in blood (see Fig. 1). Subsequently, cerebral blood flow is strongly upregulated locally by arteriolar vasodilation to replenish the tissue,

∗ Corresponding author: jaime.arias@inria.fr
‡ Univ. Grenoble Alpes, Inria, MISTIS, LJK, F-38000 Grenoble, France
§ CEA/NeuroSpin and Inria Saclay, Parietal, France
¶ Inserm, U1216, F-38000 Grenoble, France
‖ Univ. Grenoble Alpes, GIN, F-38000 Grenoble, France

increasing local blood oxygen saturation in veins and capillaries above the initial level. Oxygenated and deoxygenated blood has different magnetic properties. As a result, the above causes a BOLD signal increase. Thus, the BOLD signal is an indirect measure of cerebral activity based on physiological changes in oxygen consumption, cerebral blood flow and cerebral blood volume.
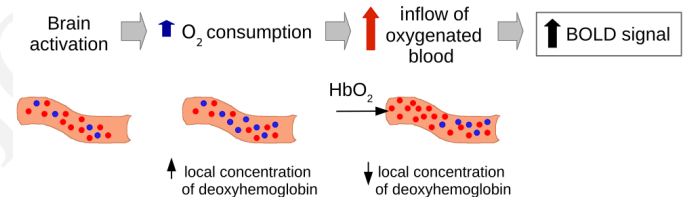

Fig. 1: fMRI BOLD signal [OLKT90]. The BOLD signal measures the local changes in blood oxygenation ocurring during brain activity.

fMRI data is acquired by repeated imaging of the brain while the subject or patient executes a task or receives a sensory stimulus during repeated epochs separated by periods of rest. This data is analyzed by correlating the measured time-varying BOLD signal in each image location with a predicted BOLD signal, obtained by convolving the known function representing the stimulus with a Hemodynamic Response Function (HRF) modeling the delay in the vascular response. Locations in the brain where this correlation is statistically significant are considered to exhibit a neuronal response to the task or stimulus, and thus to be involved in its cognitive processing.

BOLD fMRI is non-invasive, non-ionizing, and gives access *in vivo* to brain activity with a relatively high spatial resolution. However, it is highly dependent of the HRF of the brain. The BOLD signal does not give access to true physiological parameters such as cerebral blood flow or cerebral blood volume, but rather measures a mixture of these quantities that is difficult to untangle. In this regard, BOLD is a very interesting tool in neuroscience, but in general it is not widely used for clinical applications because the impact of physiopathological situation on the HRF and the response amplitude are unknown, hampering the BOLD signal interpretation. For instance, the vascular response giving rise to the BOLD signal is altered in normal ageing [FGM+14]
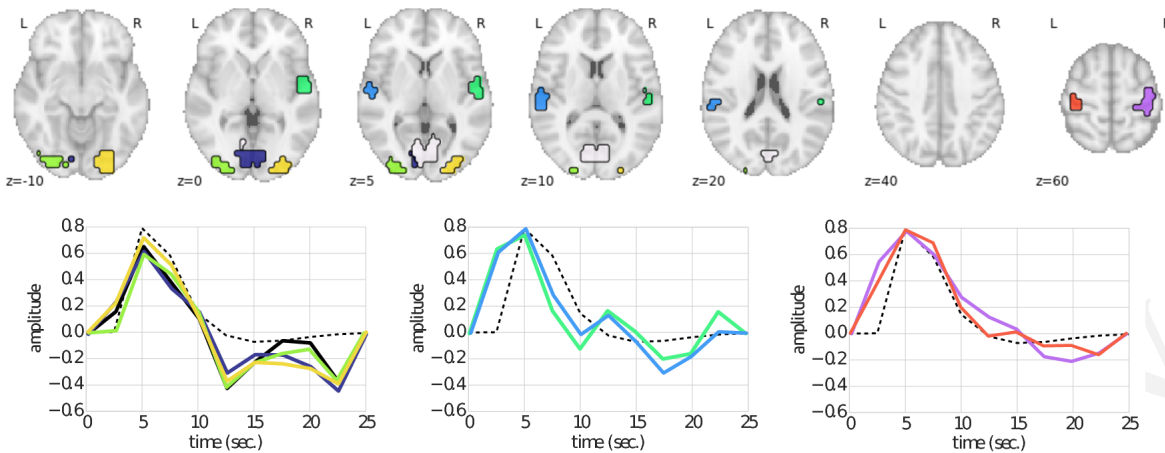
Fig. 2: *HRF computed using PyHRF from BOLD data in several parcels belonging, respectively from left to right, to visual (yellow, dark blue and green parcels), auditory (cyan and light green parcels) and motor (red and purple parcels) regions.*

and pathologies like Alzheimer's disease [CVM⁺11] or Stroke [AVT⁺14].

Most used open source libraries for the analysis of fMRI data (*e.g.,* SPM[1], FSL[2]) consider the HRF as constant in all the brain and the same for all subjects. However, several works (see [BVC13] for a survey) show that the HRF changes across different regions of the brain and across individuals, increasing thus the possibility of obtaining false negatives and decreasing the reliability of the results. The software PyHRF [VBR⁺14] was developed to overcome the above limitation by analyzing fMRI data using a Joint Detection-Estimation (JDE) approach.

In the JDE approach, the detection of the cortical activation is achieved together with the estimation of the unknown HRF response by analyzing non-smoothed data. This detection-estimation is calculated on different parcels of interest paving the cerebral volume. Therefore, PyHRF allows to navigate throughout the brain and to focus on regions of interest during the experiment in order to visualize the activations and their temporal behavior through the estimated HRF. Over the last years, efforts have been made in terms of image processing, user-friendliness and usability of the PyHRF package to make it more easy to use by non experts and clinicians.

Next, we present the PyHRF package. Then we illustrate its use on real fMRI data. Finally, we conclude by discussing directions of current/future work. An online Jupyter notebook containing the results presented here can be found at http://www.pyhrf.org/scipy2017_notebook.

PyHRF

PyHRF (http://www.pyhrf.org) is an open source tool implemented in Python that allows to jointly detect activation and estimate (JDE) the hemodynamic response function (HRF) [MIV⁺08], which gives the temporal changes in the BOLD effect induced by brain activity. This estimation is not easy in a *voxel-wise* manner [PJG⁺03], and a spatial structure was added to JDE [VRC10] in order to get reliable estimates. In this regard, HRF estimation in JDE is *parcel-wise* and an input parcellation is required. However, the use of the Markov Chain Monte Carlo

(MCMC) method for estimation added a huge computational load to the solution, leading to the development of a faster method to deal with the parameter estimation. Thus, a Variational Expectation Maximization (VEM) solution [CVF⁺13] was implemented.

JDE aims at improving activation detection by capturing the correct hemodynamics, since using the wrong HRF function could hide existing activations. The use of a canonical HRF is usually sufficient for activation detection. However, HRF functions have been found to have different shapes in different regions [HOM04], and to have different delays in specific populations [BVC13]. They change depending on pathologies such as stenosis.

Fig. 2 shows some HRF functions estimated by PyHRF from BOLD data on a healthy adult. This data was acquired in a block-design setting with visual, auditory and motor experimental conditions. The parcels correspond to regions of the brain that are known induce evoked activity in response to these experimental conditions. Observe that the HRF estimated in the visual and motor regions (first and third figure from left to right) are well approximated by the canonical HRF whereas in the auditory area (second figure from left to right), the recovered hemodynamic profiles peak earlier than the canonical shape.

Standard methods (*e.g.,* GLM), with the posterior classical statistics applied, give Statistical Parametric Maps (SPM) that describe the significance of the activation in each region. JDE is a Bayesian approach and estimates, for each parameter, posterior probability functions. For this reason, we can compute Posterior Probability Maps (PPMs) from the output of PyHRF. These PPMs are not directly comparable to the classical SPM maps, but give a similar measure of significance of activation. For instance, in Fig. 4 we show the SPM and PPM maps for a visual experimental condition in the same data used for Fig. 2. We used the package Nilearn (http://nilearn.github.io) to generate the brain maps presented in this document.

In Fig. 3 we present the inputs and the outputs of PyHRF for the analysis of BOLD data. It needs as inputs the data volume (BOLD), the experimental paradigm, and a parcellation of the brain. After running the JDE algorithm, the output will consist of HRF functions per parcel, BOLD effect maps per experimental condition, and PPMs per condition. In the next section, we will describe in more details these elements and how to use PyHRF.

---

1. SPM official website: http://www.fil.ion.ucl.ac.uk/spm/software/
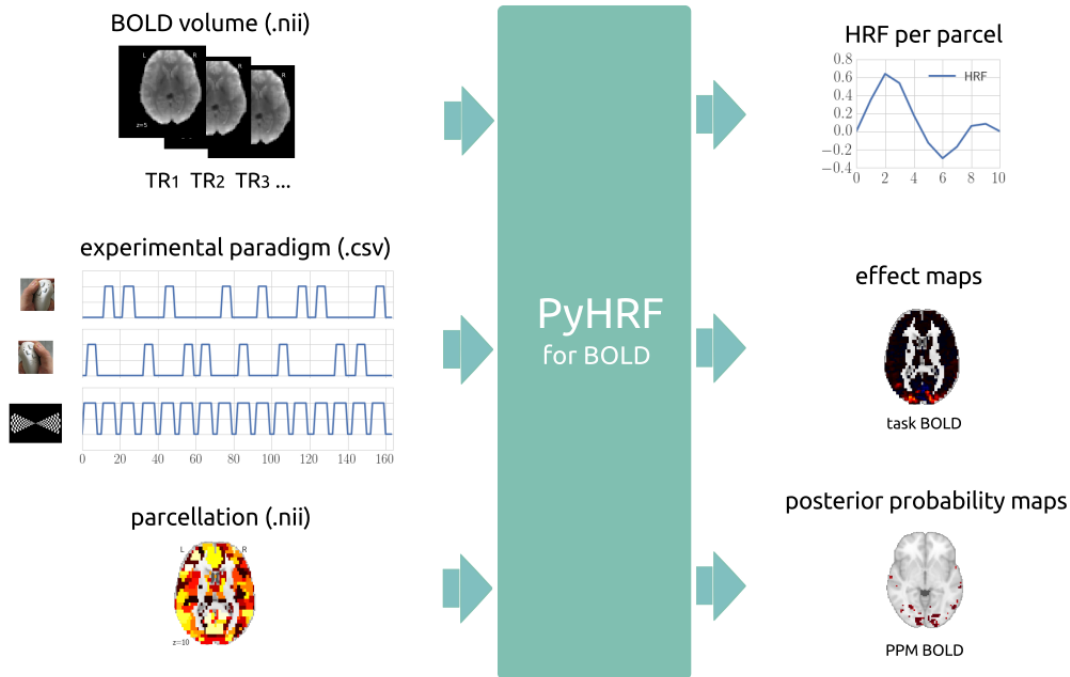2. FSL official website: https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/

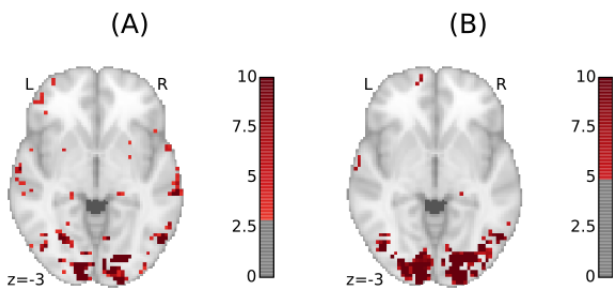**Fig. 3:** *Inputs and outputs of PyHRF when analyzing BOLD data.*

Fig. 4: *A) PPM and B) SPM maps computed with JDE and GLM, respectively. Scale is logarithmic.*

Example of Use

To illustrate the use of PyHRF, we will describe the steps for the analysis of BOLD data. A Jupyter notebook containing the complete code is available at http://www.pyhrf.org/scipy2017_notebook.

Getting fMRI BOLD Data

First of all, we need to get our fMRI BOLD data. In our running example, we will analyze the dataset used in [GSB⁺13]. This dataset (ds000114) is open shared and it can be downloaded from https://openfmri.org/dataset/ds000114/. For convenience, we implemented the method get_from_openfmri that uses the library fetchopenfmri (https://github.com/wiheto/fetchopenfmri) to download datasets from the site openfmri. For instance, the following code downloads the dataset ds000114 to the folder ~/data.

```
>>> dataset_path = get_from_openfmri('114', '~/data')
Dataset ds000114 already exists
/home/jariasal/data/openfmri/ds000114_R2.0.1
```

Briefly, in this dataset ten healthy subjects in their fifties were scanned twice using an identical experimental paradigm. This paradigm consists of five task-related fMRI time series: finger, foot and lip movement; overt verb generation; covert verb generation; overt word repetition; and landmark tasks. For the sake of simplicity, we will focus our analysis only on motor tasks (*i.e.,* finger, foot and lips movement). Fig. 5 shows the paradigm containing only the three tasks mentioned above. As we can see in the experimental paradigm, tasks do not overlap each other and stimuli are presented to the subject during a certain time (*i.e.,* block paradigm).
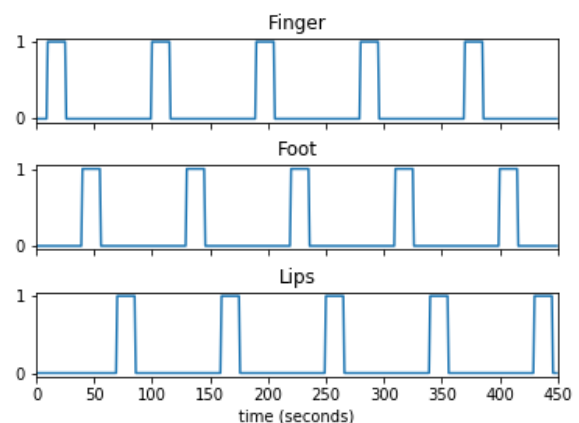

Fig. 5: *Experimental paradigm of the dataset ds000114. We show only the motor tasks of the dataset (finger, foot and lips movement).*

fMRI BOLD Preprocessing

Once we have the BOLD volumes, we need to apply some transformations to the images in order to correct for possible errors that may have occurred along the acquisition. For instance,

a BOLD volume (*e.g.,* a whole brain) is usually not built at once but using a series of successively measured 2D slices. Each slice takes some time to acquire, so slices are observed at different time points, leading to suboptimal statistical analysis.

We used the library `Nipype` (https://github.com/nipy/nipype) to define and apply our preprocessing pipeline. This library allows to use robust tools, such as SPM and FSL, in an easy manner. The proposed workflow (see Fig. 6) starts by uncompressing the images since they are in a `nii.gz` format (`gunzip_func` and `gunzip_anat` nodes). After, it applies a *slice timing* operation in order to make appear that all voxels of the BOLD volume have been acquired at the same time. We then apply a *realignment* in order to correct for head movements. Moreover, we apply a *coregistration* operation in order to have the anatomical image (high spatial resolution) in the same space as the BOLD images. Finally, we *normalize* our images in order to transform them into a standard space (a template).

Fig. 6: *Preprocessing pipeline defined with `Nipype` and used in our running example.*

The pipeline described above was run for the images of all subjects from the dataset (*i.e.,* 10 subjects) on multiple processors, since `Nipype` uses the library `joblib` (https://github.com/joblib/joblib). We used the acquisition parameters presented in [GSB+13] to parameterize each preprocessing operation. For instance, the number of slices for the volume, the time for acquiring all slices (TR), and the order in which they were acquired (*e.g.,* interleaved).

In the following snippet, we show a portion of the code to define the slice timing task with `Nipype`.

```
>>> import nipype.interfaces.spm as spm
>>> import nipype.pipeline.engine as npe

# Acquisition parameters
>>> TR = 2.5
>>> NUM_SLICES = 30
>>> TA = TR - (TR / NUM_SLICES)
>>> REF_SLICE = 1

# interleaved slice order
>>> SLICE_ORDER = list(range(1, NUM_SLICES+1, 2) +
                       range(2, NUM_SLICES+1, 2))

# slice timing with SPM
>>> spm_st = spm.SliceTiming(num_slices=NUM_SLICES,
                             time_repetition=TR,
                             time_acquisition=TA,
                             slice_order=SLICE_ORDER,
                             ref_slice=REF_SLICE)
>>> slice_timing = npe.Node(spm_st,
                            name='slice_timing_node')
```
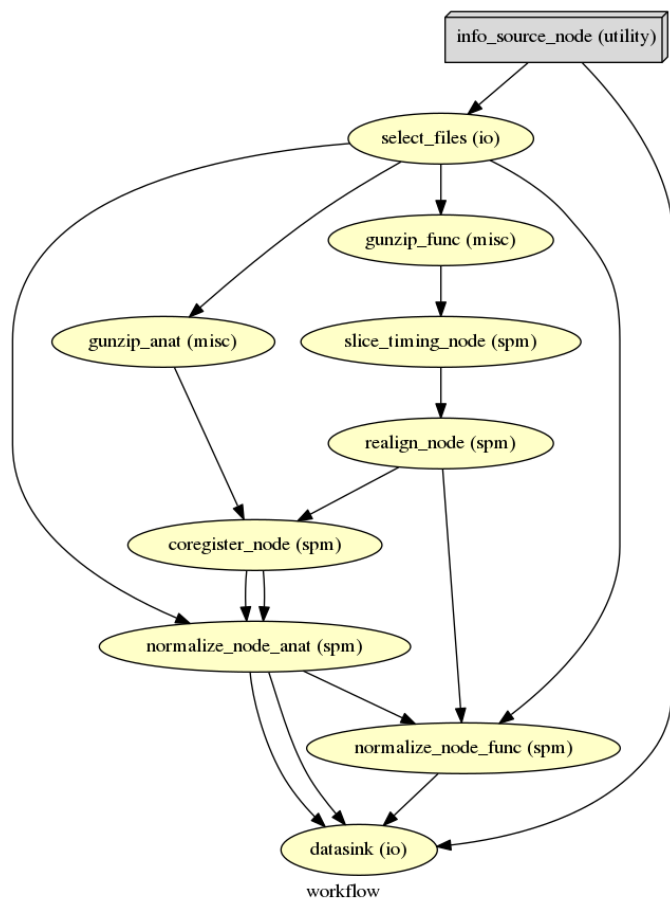
PyHRF Analysis (Inputs)

So far, we have prepared our functional and structural images for BOLD analysis. It is important to note that PyHRF receives *non-smoothed* images as input, thus we excluded this operation from our preprocessing pipeline.

For the sake of simplicity, in our running example we only analyze the 4th subject from our dataset. Additionally, we use the package `Nilearn` to load and visualize neuroimaging volumes. Fig. 7 shows the mean of the functional images of the 4th subject after preprocessing.

Fig. 7: *Mean of all preprocessed functional images (over time) of the 4th subject of the dataset `ds000114`.*

As we explained before, the JDE framework estimates HRF parcel-wise. This means that PyHRF needs a parcellation mask to perform the estimation-detection. The package provides a Willard atlas [RAM+15] (see Fig. 8) created from the files distributed by Stanford (http://findlab.stanford.edu/functional_ROIs.html). This atlas has a voxel resolution of 3x3x3 mm and a volume shape of 53x63x52 voxels.

Fig. 8: *Willard atlas [RAM+15].*

We used the method `get_willard_mask` to resize the original atlas to match the shape of the BOLD images to be

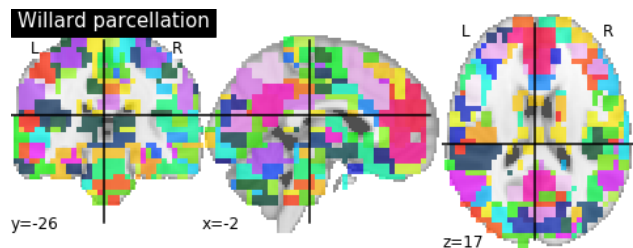| session	condition	onset	duration	amplitude
0	Finger	10	15.0	1
0	Foot	40	15.0	1
0	Lips	70	15.0	1
0	Finger	100	15.0	1
0	Foot	130	15.0	1
0	Lips	160	15.0	1
0	Finger	190	15.0	1
0	Foot	220	15.0	1
0	Lips	250	15.0	1
0	Finger	280	15.0	1
0	Foot	310	15.0	1
0	Lips	340	15.0	1
0	Finger	370	15.0	1
0	Foot	400	15.0	1
0	Lips	430	15.0	1

TABLE 1: *Experimental paradigm of the dataset* `ds000114` *containing only motor stimuli. The column organization of the file follows the PyHRF format.*

analyzed. In addition, this method saves the resampled mask in a specified path. For instance, Fig. 8 shows the Willard atlas resized to the shape of the functional image in Fig. 7. The following code illustrates how to resize the Willard atlas provided by PyHRF to match the shape of the image ~/data/bold.nii, and saves it in the folder ~/pyhrf.

```
>>> willard = get_willard_mask('~/pyhrf',
                               '~/data/bold.nii')
/home/jariasal/pyhrf/mask_parcellation/willard_3mm.nii
```

PyHRF also needs the experimental paradigm as input. It must be given as a `csv` file following the convention described in the documentation (https://pyhrf.github.io/manual/paradigm.html). For the sake of convenience, we used the method `convert_to_pyhrf_csv` to read the paradigm file provided by the dataset (a `tsv` file) and rewrite it using the PyHRF format. Since each dataset has its own format for the paradigm, we give it as an input to our method.

```
>>> columns_tsv = ['onset', 'duration', 'weight',
                   'trial_type']
>>> paradigm = convert_to_pyhrf_csv(
               '~/data/paradigm.tsv', 0,
               columns_tsv)
/tmp/tmpM3zBD5
```

Table 1 shows the experimental paradigm of the dataset `ds000114` written using the PyHRF format. Note that it only contains motor stimuli since we are only interested in them for our BOLD analysis.

PyHRF Analysis (Run)

Now we are ready to start our BOLD analysis with PyHRF. For that, we need to define some important parameters of the underlying JDE model (*e.g.,* `beta`, `hrf-hyperprior`, `sigma-h`, `drifts-type`) and a folder to save the output (`output`).

Moreover, we need to specify if we want to estimate the HRF response or use, for example, its canonical form. In our running example, we will estimate the HRF (`estimate-hrf`) with a time resolution (`dt`) of 1.25 s, a duration (`hrf-duration`) of 25.0 s, and we force to zero the beginning and ending of the response (`zero-constraint`).

Once the parameters of the model have been defined, we run our analysis by using the command-line tool `pyhrf_jde_vem_analysis` provided by PyHRF. We can execute the analysis using several processors (`parallel`) because PyHRF uses the library `joblib`. The reader can found more details about this command and its parameters in the PyHRF documentation.

```
pyhrf_jde_vem_analysis [options] TR atlas_file \
                       paradigm_file bold_images

pyhrf_jde_vem_analysis \
  --estimate-hrf \
  --dt 1.25 \
  --hrf-duration 25.0 \
  --zero-constraint \
  --beta 1.0 \
  --hrf-hyperprior 1000 \
  --sigma-h 0.1 \
  --drifts-type cos \
  --parallel \
  --log-level WARNING \
  --output /home/jariasal/pyhrf \
  2.5 \
  {$HOME}/pyhrf/mask_parcellation/willard_3mm.nii \
  /tmp/tmpM3zBD5
  {$HOME}/data/bold.nii
```

PyHRF Analysis (Output)

We show in Fig. 9 the active parcels (A), the PPMs (B), and the estimated HRFs (C) generated by PyHRF for the motor task `Finger`. Reading the description given in [GSB+13], this task corresponds to finger tapping. Recall that PyHRF estimates a HRF for each parcel and generates a PPM for each condition.

We compared the output of PyHRF with the thresholded statistical maps shared on the site *Neurovault* (http://www.neurovault.org/images/307/) for the same dataset and same task (see Fig. 9). While the experimental paradigm is not optimized for JDE (standard block paradigm is not ideal to estimate different points of the HRF course), we obtained similar results to standard statistical analysis additionally providing the form of the HRF. As we can observe, at cut *z=60* both results (Fig. 9 B and D) are quite similar, showing an activation in the *supplementary motor area* and the *left primary sensorimotor cortex*.

Concluding Remarks

In this paper we presented PyHRF, a software to analyze fMRI data using a Joint Detection-Estimation (JDE) approach of the cerebral activity. It jointly detects cortical activation and estimates the Hemodynamic Response Function (HRF), in contrast to existing tools, that consider the HRF as constant over the brain and over subjects, thus aiming to improve the reliability of the results.

PyHRF is an open source software that has evolved rapidly over the last few years. As we showed, it allows to generate Posterior Probability Maps (PPMs) to describe the significance of the activation in each region of the brain. Moreover, PyHRF uses efficient estimation methods in order to provide a fast tool. Currently, the package does not provide finely tuned values for the parameters of the JDE model, leaving the user the cumbersome task of finding the best values for the estimation.

Since 2013, PyHRF has started to evolve to deal with functional Arterial Spin Labelling (fASL) [VWV+13] data, including a physiological prior to make the perfusion estimation more robust [FPVS+14] [FPFC15b]. A fast solution for fASL based on VEM was proposed in [FPFC15c], with similar results to the solution
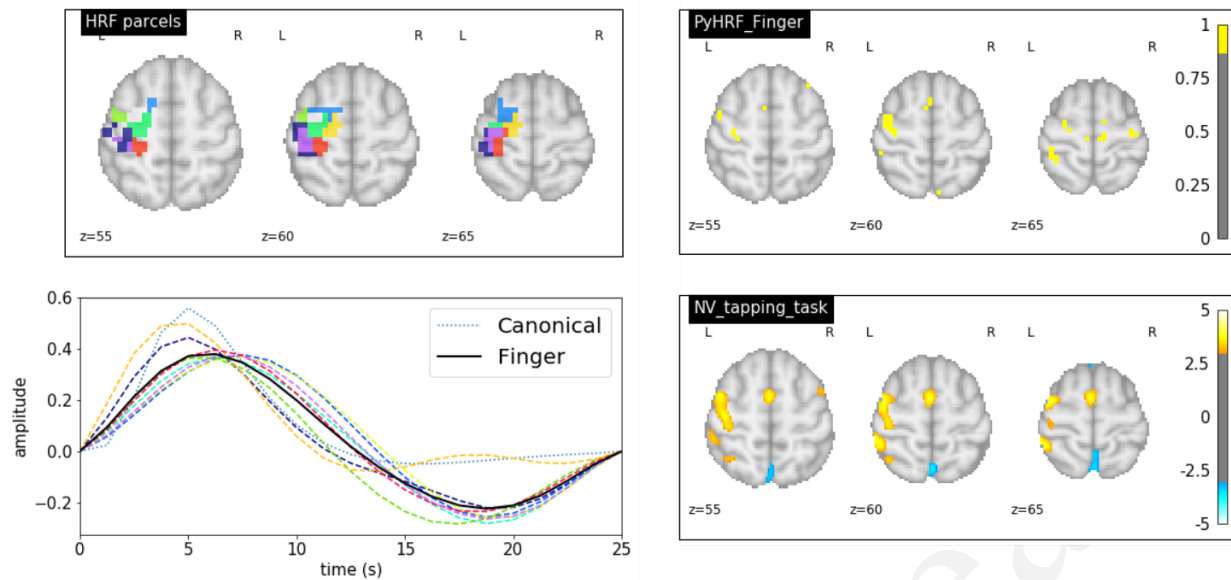
Fig. 9: *A) Active parcels, B) PPMs, and C) estimated HRFs generated by PyHRF for the dataset* `ds000114` *and the finger tapping task. D) Shows the thresholded statistical maps shared on the site* `Neurovault` *for the same dataset and task. The cut z=60 shows a high activation in the* supplementary motor area *and the* left primary sensorimotor cortex.

based on stochastic simulation techniques [FPFC15a]. Moreover, many efforts have been made in terms of image processing, user-friendliness and usability of the PyHRF tool to make it more easy to use by non experts and clinicians.

In the years to come, we plan to develop a light viewer to explore the results of PyHRF interactively. Moreover, we aspire to make the package compatible with Python 3 and extend its use to the analysis of fMRI data on small animals.

REFERENCES

[AVT+14]   A. Attyé, M. Villien, F. Tahon, J. Warnking, O. Detante, and A. Krainik. Normalization of cerebral vasoreactivity using BOLD MRI after intravascular stenting. *Human Brain Mapping*, 35(4):1320–1324, apr 2014.

[BVC13]   S. Badillo, T. Vincent, and P. Ciuciu. Group-level impacts of within- and between-subject hemodynamic variability in fMRI. *NeuroImage*, 82:433–448, nov 2013.

[CVF+13]   L. Chaari, T. Vincent, F. Forbes, M. Dojat, and P. Ciuciu. Fast joint detection-estimation of evoked brain activity in event-related fMRI using a variational approach. *IEEE Trans. on Medical Imaging*, 32(5):821–837, May 2013.

[CVM+11]   S. Cantin, M. Villien, O. Moreaud, I. Tropres, S. Keignart, E. Chipon, J.-F. Le Bas, J. Warnking, and A. Krainik. Impaired cerebral vasoreactivity to CO2 in Alzheimer's disease using BOLD fMRI. *NeuroImage*, 58(2):579–587, sep 2011.

[FGM+14]   M. Fabiani, B. A. Gordon, E. L. Maclin, M. A. Pearson, C. R. Brumback-Peltz, K. A. Low, E. McAuley, B. P. Sutton, A. F. Kramer, and G. Gratton. Neurovascular coupling in normal aging: A combined optical, ERP and fMRI study. *NeuroImage*, 85:592–607, jan 2014.

[FPFC15a]   A. Frau-Pascual, F. Forbes, and P. Ciuciu. Comparison of stochastic and variational solutions to ASL fMRI data analysis. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 85–92. Springer, 2015.

[FPFC15b]   A. Frau-Pascual, F. Forbes, and P. Ciuciu. Physiological models comparison for the analysis of ASL fMRI data. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pages 1348–1351. IEEE, 2015.

[FPFC15c]   A. Frau-Pascual, F. Forbes, and P. Ciuciu. Variational physiologically informed solution to hemodynamic and perfusion response estimation from ASL fMRI data. In *Pattern Recognition in NeuroImaging (PRNI), 2015 International Workshop on*, pages 57–60. IEEE, 2015.

[FPVS+14]   A. Frau-Pascual, T. Vincent, J. Sloboda, P. Ciuciu, and F. Forbes. Physiologically informed Bayesian analysis of ASL fMRI data. In *Bayesian and grAphical Models for Biomedical Imaging*, pages 37–48. Springer, 2014.

[GSB+13]   K. J. Gorgolewski, A. Storkey, M. E. Bastin, I. R. Whittle, J. M. Wardlaw, and C. R. Pernet. A test-retest fMRI dataset for motor, language and spatial attention functions. *GigaScience*, 2(1):6, 2013.

[HOM04]   D. A. Handwerker, J. M. Ollinger, and D. Mark. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *Neuroimage*, 21:1639–1651, 2004.

[MIV+08]   S. Makni, J. Idier, T. Vincent, B. Thirion, G. Dehaene-Lambertz, and P. Ciuciu. A fully Bayesian approach to the parcel-based detection-estimation of brain activity in fMRI. *Neuroimage*, 41(3):941–969, 2008.

[OLKT90]   S. Ogawa, T. M. Lee, A. R. Kay, and D. W. Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872, dec 1990.

[PJG+03]   P. Ciuciu, J.-B. Poline, G. Marrelec, J. Idier, Ch. Pallier, and H. Benali. Unsupervised robust non-parametric estimation of the hemodynamic response function for any fMRI experiment. *IEEE Trans. Med. Imag.*, 22(10):1235–1251, Oct. 2003.

[RAM+15]   J. Richiardi, A. Altmann, A.-C. Milazzo, C. Chang, M. M. Chakravarty, T. Banaschewski, G. J. Barker, A. L. W. Bokde, U. Bromberg, C. Buchel, P. Conrod, M. Fauth-Buhler, H. Flor, V. Frouin, J. Gallinat, H. Garavan, P. Gowland, A. Heinz, H. Lemaitre, K. F. Mann, J.-L. Martinot, F. Nees, T. Paus, Z. Pausova, M. Rietschel, T. W. Robbins, M. N. Smolka, R. Spanagel, A. Strohle, G. Schumann, M. Hawrylycz, J.-B. Poline, M. D. Greicius, L. Albrecht, C. Andrew, M. Arroyo, E. Artiges, S. Aydin, C. Bach, T. Banaschewski, A. Barbot, G. Barker, N. Boddaert, A. Bokde, Z. Bricaud, U. Bromberg, R. Bruehl, C. Buchel, A. Cachia, A. Cattrell, P. Conrod, P. Constant, J. Dalley, B. Decideur, S. Desrivieres, T. Fadai, H. Flor, V. Frouin, J. Gallinat, H. Garavan, F. G. Briand, P. Gowland, B. Heinrichs, A. Heinz, N. Heym, T. Hubner, J. Ireland, B. Ittermann, T. Jia, M. Lathrop, D. Lanzerath, C. Lawrence, H. Lemaitre, K. Ludemann, C. Macare, C. Mallik, J.-F. Mangin, K. Mann, J.-L. Martinot, E. Mennigen, F. Mesquita de Carvahlo, X. Mignon, R. Miranda, K. Muller, F. Nees, C. Nymberg, M.-L. Paillere, T. Paus, Z. Pausova, J.-B. Poline, L. Poustka, M. Rapp, G. Robert, J. Reuter, M. Rietschel, S. Ripke, T. Robbins, S. Rodehacke, J. Rogers, A. Romanowski, B. Ruggeri, C. Schmal, D. Schmidt, S. Schneider, M. Schumann, F. Schubert,

Y. Schwartz, M. Smolka, W. Sommer, R. Spanagel, C. Speiser, T. Spranger, A. Stedman, S. Steiner, D. Stephens, N. Strache, A. Strohle, M. Struve, N. Subramaniam, L. Topper, R. Whelan, S. Williams, J. Yacubian, M. Zilbovicius, C. P. Wong, S. Lubbe, L. Martinez-Medina, A. Fernandes, and A. Tahmasebi. Correlated gene expression supports synchronous activity in brain networks. *Science*, 348(6240):1241–1244, jun 2015.

[VBR⁺14] T. Vincent, S. Badillo, L. Risser, L. Chaari, C. Bakhous, F. Forbes, and P. Ciuciu. Flexible multivariate hemodynamics fMRI data analyses and simulations with PyHRF. *Frontiers in Neuroscience*, 8, apr 2014.

[VRC10]   T. Vincent, L. Risser, and P. Ciuciu. Spatially adaptive mixture modeling for analysis of within-subject fMRI time series. *IEEE Trans. on Medical Imaging*, 29(4):1059–1074, April 2010.

[VWV⁺13] T. Vincent, J. Warnking, M. Villien, A. Krainik, P. Ciuciu, and F. Forbes. Bayesian Joint Detection-Estimation of cerebral vasoreactivity from ASL fMRI data. In *16th Proc. MICCAI*, volume 2, pages 616–623, Nagoya, Japan, September 2013.