

# Pyhrf: a package to extract and study hemodynamics from fMRI data

Thomas Vincent<sup>1</sup>, Philippe Ciuciu<sup>1</sup>, Lotfi Chaari<sup>2</sup>, and Solveig Badillo<sup>1</sup>

<sup>1</sup>CEA/DSV/I<sup>2</sup>BM/Neurospin, LNAO, Gif-Sur-Yvette, France

<sup>2</sup>INRIA, MISTIS, Grenoble, France

Pyhrf is a set of tools for within-subject fMRI data analysis, focused on the characterisation of the hemodynamics. More precisely, this software addresses the two main tasks involved in fMRI analysis: (i) detect and localize which cerebral regions are activated by a given experimental paradigm and (ii) estimate the underlying dynamics of activation by recovering the so-called Hemodynamic Response Function (HRF). The main outputs are then a set of 3D statistical maps of cerebral activations along with the time-series describing the HRFs for the set of all brain regions (4D volume). The analysis can also be performed on the cortical surface from projected BOLD signals and then produces functional textures to be displayed on the input cortical mesh. To this end, pyhrf implements two different approaches: a voxel-wise and condition-wise HRF estimation [1] and a parcel-wise spatially adaptive joint detection-estimation algorithm [2,3]. This tool provides interesting perspectives so as to understand the differences in the HRFs of different populations (infants, children, adults, patients ...). Within the treatment pipeline of an fMRI analysis, `pyhrf` steps in after data preprocessings (slice-timing, realignment, normalization).

`pyhrf` is mainly written in Python, with some C-extension that handle computation intensive parts of the algorithms. The package relies on classical scientific libraries: `numpy`, `scipy`, `matplotlib` as well as `Nibabel` to handle input/outputs and `NiPy` which provides tools for functional data analysis. `pyhrf` can be used in a stand-alone fashion and provides a set of simple commands in a modular fashion. The setup process is handled through XML files which can be adapted by the user from a set of templates. This format was chosen for its hierarchical organisation which suits the nested nature of the algorithm parametrisations. A dedicated XML editor is provided with a `PyQt` graphical interface for a quicker edition and also a better review of the treatment parameters. When such an XML setup file is generated *ab initio*, it defines a default analysis which involves a small real data set shipped with the package. This allows for a quick testing of the algorithm and is also used for demonstration purpose. In the same respect, an artificial fMRI data generator is provided where the user can test the behaviour of the algorithms with different activation configurations, HRF shapes, nuisance types (noise, low frequency drifts) and paradigms (slow/fast event-related or bloc designs). Concerning the analysis process, which can be computation intensive, `pyhrf` handles parallel computing through the python software `soma-workflow` for the exploitation of cluster units as well as multiple cores computers. Finally, results can be browsed by a dedicated viewer based on `PyQt` and `matplotlib` which handles n-dimensionnal images and provide suitable features for the exploration of whole brain hemodynamical results.

## References

- [1] 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.*, vol. 22, no. 10, pp. 1235–1251, oct. 2003.
- [2] T. Vincent, L. Risser, J. Idier, and P. Ciuciu, “Spatially adaptive mixture modelling for analysis of fMRI time series,” in *Proc. 15th HBM*, San Francisco, CA, USA, juin 2009.
- [3] L. Chari, F. Forbes, P. Ciuciu, T. Vincent, and Michel Dojat, “Bayesian variational approximation for the joint detection-estimation of brain activity in fMRI,” Tech. Rep., INRIA Rhône-Alpes and LNAO/NeuroSpin, Grenoble, France, fév. 2011.