



HAL
open science

Flexible multivariate hemodynamics fMRI data analyses and simulations with PyHRF

Thomas Vincent, Solveig Badillo, Laurent Risser, Lotfi Chaari, Christine Bakhous, Florence Forbes, Philippe Ciuciu

► To cite this version:

Thomas Vincent, Solveig Badillo, Laurent Risser, Lotfi Chaari, Christine Bakhous, et al.. Flexible multivariate hemodynamics fMRI data analyses and simulations with PyHRF. *Frontiers in Neuroscience*, 2014, Brain Imaging methods (eCollection 2014), 8 (Article 67), pp.23. 10.3389/fnins.2014.00067 . hal-01084249

HAL Id: hal-01084249

<https://inria.hal.science/hal-01084249v1>

Submitted on 18 Nov 2014

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

1

Flexible multivariate hemodynamics fMRI data analyses and simulations with PyHRF

Thomas Vincent^{1,2}, Solveig Badillo², Laurent Risser^{2,3}, Lotfi Chaari^{1,4},
Christine Bakhous¹, Florence Forbes¹ and Philippe Ciuciu^{2,*}

¹INRIA, MISTIS, Grenoble University, LJK, Grenoble, France

²CEA/DSV/I²BM NeuroSpin center, Bât. 145, F-91191 Gif-sur-Yvette, France

³CNRS, Toulouse Mathematics Institute, UMR 5219, Toulouse, France

⁴INP-ENSEEIH / CNRS UMR 5505, University of Toulouse, IRIT, Toulouse, France

Correspondence*:

Philippe Ciuciu

CEA/DSV/I²BM NeuroSpin center, Bât. 145, F-91191 Gif-sur-Yvette, France,
philippe.ciuciu@cea.fr

Research Topic

2 ABSTRACT

3 As part of fMRI data analysis, the `pyhrf` package provides a set of tools for addressing the two
4 main issues involved in intra-subject fMRI data analysis: (i) the localization of cerebral regions
5 that elicit evoked activity and (ii) the estimation of the activation dynamics also referenced to
6 as the recovery of the Hemodynamic Response Function (HRF). To tackle these two problems,
7 `pyhrf` implements the Joint Detection-Estimation framework (JDE) which recovers parcel-level
8 HRFs and embeds an adaptive spatio-temporal regularization scheme of activation maps. With
9 respect to the sole detection issue (i), the classical voxelwise GLM procedure is also available
10 through `nipy`, whereas Finite Impulse Response (FIR) and temporally regularized FIR models
11 are implemented to deal with HRF estimation concerns (ii). Several parcellation tools are also
12 integrated such as spatial and functional clustering. Parcellations may be used for spatial
13 averaging prior to FIR/RFIR analysis or to specify the spatial support of the HRF estimates
14 in the JDE approach. These analysis procedures can be applied either to volumic data sets or
15 to data projected onto the cortical surface. For validation purpose, this package is shipped with
16 artificial and real fMRI data sets, which are used in this paper to compare the outcome of the
17 different available approaches. The artificial fMRI data generator is also described to illustrate
18 how to simulate different activation configurations, HRF shapes or nuisance components. To
19 cope with the high computational needs for inference, `pyhrf` handles distributing computing
20 by exploiting cluster units as well as multiple cores computers. Finally, a dedicated viewer is
21 presented, which handles n -dimensional images and provides suitable features to explore whole
22 brain hemodynamics (time series, maps, ROI mask overlay).

23 **Keywords:** medical imaging analysis, fMRI, Bayesian inference, python, scientific computing

1 INTRODUCTION

24 As Magnetic Resonance Imaging (MRI) is a growing imaging modality in neuroscience, the need
25 for powerful tools to explore the increasing amount of data is more and more significant. This data
26 growth is quantitative as cohort sizes are getting bigger through the development of international multi-
27 centre projects like the Human Brain Project **Koslow and Huerta** (2013) but also qualitative as high
28 field magnets become more and more available **Duyn and Koretsky** (2011). Functional MRI (fMRI)
29 especially benefits from these improvements and the experimenter has access to finer spatial (~ 1 mm)
30 and temporal (~ 1 sec.) resolutions and also higher signal-to-noise ratio (SNR). In particular, the higher
31 temporal resolution combined with higher SNR allows a better recovery of dynamical processes so
32 that we no longer have to accommodate with only static mappings of cerebral activity. In this context,
33 `pyhrf` aims at extracting dynamical features from fMRI data and especially the Blood Oxygenated
34 Level Dependent (BOLD) modality (**Ogawa et al.** (1990)). **The observed BOLD signal is an indirect**
35 **measure of the neural activity via the oxygen variation induced by the neuro-vascular coupling. Therefore,**
36 **analysis methods have to formalize a hemodynamic model in order to make inference on neural processes.**
37 **However, even if BOLD variations are known to correlate with neural activity, it is difficult to disentangle**
38 **the neural and the vascular components in terms of dynamics.** As the employed methodology mainly
39 resorts to linear systems, dynamical processes are summarized within the so-called Hemodynamic
40 Response Function (HRF), which is the impulse response that links neuronal stimulation to the fMRI
41 signal, through the neuro-vascular coupling. In fact, the package offers various tools to analyze evoked
42 fMRI data ranging from spatial mappings such as those provided by the General Linear Model (GLM)
43 framework (**Friston et al.** (1995)) to finer hemodynamics models as provided by the joint detection-
44 estimation (JDE) approach described in **Makni et al.** (2005, 2008); **Vincent et al.** (2010). **Through**
45 **a bilinear and time-invariant system, the JDE approach models an unknown Hemodynamic Response**
46 **Function (HRF) at the level of a group of voxels (termed a parcel in the following) as well as voxel-**
47 **and condition-specific response levels to encode the local magnitudes of this response. The HRF is only**
48 **constrained to be smooth (temporal regularization) and can cover a wide variety of shapes. The response**
49 **levels are spatially regularized within each parcel. Hence, the JDE approach is a spatially adaptive GLM**
50 **built on an unknown parcel-dependent HRF with spatio-temporal regularization.**

51 The usage of each tool amounts to a model choice which is driven by the features required by the
52 experimenter's questioning. If one is only interested in obtaining classical detection results where the
53 canonical HRF embodies a standard and widely recognized choice, a GLM based on this canonical
54 HRF (and possibly its temporal derivatives) may be sufficient. Indeed, even if the between-region
55 hemodynamics variability is acknowledged, the canonical HRF can provide good results in regions
56 where it has precisely been calibrated such as temporal and occipital cortices as studied by **Boynton**
57 **et al.** (1996). However, if one is interested in detecting activation in regions involving more complex
58 processes or where potential hemodynamics delays happen due to varying reaction delays or pathological
59 cases, hemodynamic fluctuations influencing detection activation may occur that are not caught by the
60 HRF derivatives or function bases. Moreover, if one is interested in studying the dynamics features of
61 the response, an *explicit* HRF estimation is required. The main question in this case concerns the need
62 for condition-specific features or not, namely for an HRF estimation associated with each experimental
63 condition or for a single HRF estimate associated with all conditions, as proposed in the JDE framework.
64 If explicit condition-wise HRFs are required, the best methodological tool to use is the temporally
65 Regularized FIR (RFIR) developed in **Marrelec et al.** (2003); **Ciuciu et al.** (2003). Otherwise, if
66 variability is expected only across separated and specialized regions, the JDE framework is well-suited.
67 Indeed, within a specialized region, if only one condition exhibits activity then the region-specific HRF
68 can be considered a condition-specific HRF. The performance of RFIR models depends nonetheless on
69 the number of experimental conditions involved in the paradigm because the higher this number, the larger
70 the number of parameters to estimate and thus the fewer the number of degrees of freedom for statistical
71 testing. The model choice depends thus also on the experimental paradigm. First, it is worth noticing
72 that the use of the JDE formulation is **less relevant** to analyze block paradigm data since the signal

73 variability in this case is hardly significant. The JDE formalism is actually more adapted to fast event-
74 related paradigms or to paradigms including many conditions, like the localizer paradigm (10 conditions)
75 introduced by **Pinel et al.** (2007) and used hereafter in this paper. The JDE approach is also optimally
76 tuned to combined analysis of hemodynamics features with the detection of activated brain areas. To sum
77 up on the model choice, the JDE model provides a fair compromise with the possibility for the user to
78 adapt the model to the studied region.

79 With respect to the sole detection aspect, JDE also delivers interesting and complementary results for
80 activation detection compared to classical GLM. It is worth noticing that spatial regularization, which
81 is necessary due to the low SNR in fMRI, is not enforced in the same way between methods. In the
82 GLM, FIR and RFIR cases, there is no embedded spatial regularization within models. Indeed, the data
83 are usually spatially smoothed with a fixed Gaussian kernel as part of preprocessings. In contrast, in
84 the JDE case, spatial correlation is embedded through hidden Markov models. The amount of spatial
85 correlation is automatically tuned and also adaptive across brain regions, therefore avoiding any prior
86 invariant smoothing. As regularization methods require important computational load, a more efficient
87 variational inference scheme have been developed inside the JDE framework in **Chari et al.** (2013). Of
88 course, the computational cost is less and less a limiting factor with the increase of CPU power and the
89 advent of clusters, parallel processing units and GPGPU. The JDE approach, available in `pyhrf`, has
90 been thought of to be used in daily routine applications with parallel computing resources.

91 `pyhrf` is mainly written in python with some C code to cope with computationally demanding parts of
92 algorithms. Originally, seminal versions of the implemented methodological tools were available in the
93 matlab HRF toolbox for SPM2. The choice to move to python has been motivated by its free access and its
94 growing dynamical scientific community mainly carried by the `scipy` project. Moreover, compared with
95 matlab, python provides easier prototyping features when it comes to building user interfaces (command
96 lines, GUIs) or linking code from other languages. Finally, this python choice has been made possible
97 thanks to the `nipy` project and especially `nibabel` to handle data reading/writing in the NIFTI format.

98 In terms of package maturity, `pyhrf` is a research tool which receives various methodological advances
99 relative to fMRI data analysis. However, `pyhrf` has the ambition to target cognitive neuroscientists
100 and clinicians so that efforts are made in terms of user-friendliness. Hence, the design is a trade-off
101 between *mutability* which is required by methodological research where specifications change frequently
102 and *usability* where user interfaces should be as stable as possible to ease external non-developer use
103 cases.

104 The rest of the paper is organized as follows. First, methods available in the package are presented,
105 comprising parcellation and detection/estimation analyses. Then, the workflow and design of the `pyhrf`
106 package are detailed which cover the user interface and code snippets for the main analysis treatments,
107 simulation framework, distributed computations and data viewer. Results illustrate the outcome of
108 geometrical and functional parcellation and their impact on detection/estimation treatments. Finally,
109 conclusions are drawn and perspective concerning future developments are foreseen.

2 METHODS

110 The main fMRI data analysis methods available in `pyhrf` are of two kinds: (i) parcellation tools that
111 segment the brain into disjoint sets of positions and (ii) activation detection / HRF estimation tools
112 that highlight correlations between the input experimental paradigm and variations in the measured
113 fMRI signal. The first kind comprises two spatial parcellation tools: Voronoi-based random parcellation,
114 as reviewed by **Aurenhammer and Klein** (2000) and balanced partitioning, developed in **Elor and**
115 **Bruckstein** (2009). The second kind comprises the GLM introduced in **Friston** (1998), the FIR model
116 described in **Henson et al.** (2000), the RFIR model developed in **Marrelec et al.** (2003); **Ciuciu et al.**
117 (2003) and the JDE approach presented in **Vincent et al.** (2010); **Risser et al.** (2011). The GLM and
118 FIR procedures are provided by `nipy` while RFIR and JDE are originally implemented in `pyhrf`. For

119 all these methods, we refer to their respective bibliographical references for an extensive presentation
 120 of their methodology. Nonetheless, the main aspects of these methods are summarized in what follows
 121 with the concern of allowing the comparison between them, especially in terms of model structure and
 122 assumptions.

123 After some details about notation and conventions, we first introduce detection/estimation methods,
 124 namely GLM, FIR and RFIR, which require the measured fMRI signal as input and the timing of the
 125 experimental paradigm. After setting the generative model common to all detection/estimation methods
 126 and a brief comparative overview, each approach is presented in more details. Subsequently, parcellation
 127 methods are presented. Spatial parcellation approaches can be applied directly to the input fMRI data and
 128 only depend on its geometry. Functional parcellation, which is a clustering of GLM results, is detailed
 129 afterwards.

2.1 NOTATION

130 *Conventions* We denote vectors with bold lower case (e.g., \mathbf{y}) and matrices with bold upper case letters
 131 (e.g., \mathbf{P}). A vector is by convention a column vector. Scalars are denoted with non-bold lower case letters
 132 (e.g., a). The transpose operation is denoted by t . Probability distribution functions (pdf) are denoted
 133 using calligraphic letters (eg, \mathcal{N} and \mathcal{G} for the Gaussian and gamma distributions). Sequence of integers
 134 are denoted as $j = 1 : J$ to indicate a range between 1 and J .

135 *Data geometry* As methods can be applied to data defined in the volume or on the cortical surface, the
 136 generic term “position” will be used in place of “voxel” (volume unit) or “node” (surface mesh unit).
 137 Position indexes are denoted j ($j = 1 : J$). Data are assumed to be masked to only keep positions within
 138 the brain. J is the total number of positions within the functional mask. In addition, when considering
 139 parcellated data, this functional mask is divided into a set of Γ parcels denoted $\mathbb{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_\gamma, \dots, \mathcal{P}_\Gamma\}$,
 140 \mathcal{P}_γ is the set of position indexes belonging to parcel γ . Let us denote $J_\gamma = |\mathcal{P}_\gamma|$ the number of positions
 141 in parcel γ .

142 *Functional data* We consider the usual case of evoked fMRI data analysis where the experimental
 143 paradigm comprising M conditions is known. The signal measured at each time of repetition (TR)
 144 is denoted $\mathbf{y}_j = \{y_{j,n}\}_{n=1:N}$ where N is the number of scans. Stimulus timing onsets for a given
 145 experimental condition $m = 1 : M$ are encoded by variable \mathbf{x}^m so that $x_t^m = 1$ if a stimulus occurs
 146 at time t up to a time step Δt , else $x_t^m = 0$. The time step is such that $\Delta t \leq TR$ and depends on the actual
 147 temporal resolution sought by the analysis method.

2.2 DETECTION/ESTIMATION METHODS

For ease of comparison, the presentation of all methods is immersed in the same formalism where the
 signal is assumed generated by a linear and time-invariant (convolution) system with additive noise. We
 also consider the usual case of taking into account a position-specific low frequency drift in the data which
 is a well known fMRI artifact produced by the aliasing of respiratory and cardiac rhythms into the low
 frequencies as studied in Yan et al. (2009). The generic forward model, reads:

$$\mathbf{y}_j = \sum_{m=1}^M \mathbf{X}^m \phi_h^m + \mathbf{P}\ell_j + \mathbf{b}_j, \quad (1)$$

148 where:

- 149 • \mathbf{P} is a fixed orthonormal basis that takes a potential drift and any other nuisance effect (e.g., motion
 150 parameters) into account. The low-frequency drift can classically be either polynomial with an order
 151 up to 5 or cosine with a cut-off of 0.01Hz,

- 152 • ℓ_j are the unknown regression weights associated to \mathbf{P} ,
- 153 • \mathbf{b}_j is the noise component,
- 154 • ϕ_h^m is a “generic” hemodynamic filter of size D whose definition varies across methods. In the GLM
- 155 framework, ϕ_h^m can be fixed to the canonical HRF or parametric when resorting to function bases
- 156 and we will note R the number of unknown parameters. In non-parametric approaches, all HRF
- 157 coefficients are estimated as in RFIR or JDE approaches,
- 158 • \mathbf{X}^m is the $N \times D$ stimulus occurrence matrix consisting of the lagged stimulus covariates for the
- 159 experimental condition m : $\mathbf{X}^m = [\mathbf{x}_{t_1}^m, \dots, \mathbf{x}_{t_N}^m]^t$ with $\mathbf{x}_{t_n}^m = (x_{t_n-d\Delta t}^m)_{0 \leq d \leq D}^t$,
- 160 • $\sum_{m=1}^M \mathbf{X}^m \phi_h^m$ is hence the summation of all stimulus-induced signal components which are
- 161 generated as the convolution between the paradigm encoded in \mathbf{X}^m and the hemodynamic filters
- 162 ϕ_h^m .

163 For the sake of simplicity, multiple-run data are not considered here but all implemented methods can

164 handle such data with a fixed-effect model (same effect size across runs), a homoscedastic noise model

165 (one noise variance for all runs) and run-specific drift coefficients.

166 To give a first overview of how this generative model structure is derived in the different approaches,

167 Table 1 provides a comparison in terms of regularization, number of unknowns and analysis duration.

168 Embedded spatial regularization is only available in the JDE procedure, while temporal regularization

169 is available in RFIR and JDE (Table 1 – 1st, 2nd rows). In terms of constraint applied to the HRF

170 shape (Table 1 – 3rd row), the basis set GLM (BS GLM) is the most constraining and the shape captured

171 depends on the choice of the function basis. In the FIR, RFIR and JDE cases, any form of HRF shape

172 can be recovered, provided that they are smooth in the case of RFIR and JDE. On Table 1 – 4th row,

173 the information on the number of unknowns conveys the level of parsimony of a given model. BS GLM,

174 FIR and RFIR have increasing model complexity as the number of parameters for the HRF increases. In

175 contrast, JDE achieves larger parsimony by making the number of unknowns associated with the HRF

176 dependent on the number of parcels rather than on the number of positions. When computing the ratio

177 between the number of unknowns and the number of data points for a typical fMRI experiment (Table 1

178 – 5th row), it appears that JDE is comparable to a GLM with derivatives. The RFIR presents the worst

179 situation with 3 times more unknowns than data points. In terms of analysis duration (Table 1 – last

180 row), GLM methods are almost instantaneous as their inference is straightforward. RFIR relies on an

181 iterative scheme to perform unsupervised estimation of the amount of temporal regularization and is

182 hence much slower. In addition, the implementation of RFIR is done in pure python with a main loop

183 over positions which worsen its slow computation speed (~ 30 h. for a whole brain analysis)¹. Therefore,

184 this approach is rather limited to the processing of some regions of interest where we expect cerebral

185 activity instead of whole brain data analysis. The computation speed of JDE is also slow, but to a lesser

186 extent as results can be obtained overnight (~ 8 h. for a whole brain analysis) on a single processing unit.

187 All these considerations on speed have to be nuanced with the access to increasing computing power and

188 distributed computations, as will be seen in section 3.3.

2.2.1 *basis set General Linear Model*

In any position j of the brain, the **basis set GLM** allows (BS GLM) for some limited hemodynamic fluctuations by modeling the hemodynamic filter function ϕ_h in Eq. (1) as a weighted sum of the fixed canonical HRF denoted h_c and its first and second order derivative h'_c, h''_c as proposed in **Friston** (1998).

¹ Note that the RFIR approach with supervised regularization is much faster with an analysis duration of 20 min. since the maximum a posteriori estimator admits a closed form expression.

Table 1. Comparative overview for all detection/estimation analysis procedures available in `pyhrf` in terms of model structure and analysis duration. “2nd order deriv.” stands for a penalization on the energy of the HRF which penalizes abrupt shape changes. The number of nuisance parameters was considered the same for all models, so that only the modeling of the stimulus-induced component is relevant to assess model parsimony. The ratio “unknowns / data” is given for a typical fMRI data analysis with $J = 4 \times 10^4$, $R = 3$, $D = 40$, $M = 10$, $\Gamma = 400$ and $N = 128$ (total number of data points: $N \times J$). The analysis duration is for a whole brain data treatment on an Intel Core i5 (M480 2.67Ghz).

	BS GLM smoothing	FIR GLM smoothing	RFIR smoothing	JDE adaptive
Spatial regularization	smoothing	smoothing	smoothing	adaptive
Temporal regularization	none	none	2 nd order deriv.	2 nd order deriv.
HRF shape constraint	function basis	free	smooth	smooth
Number of unknowns for the stimulus-induced component	$J \times R \times M$ $1 \leq R \leq 3$	$J \times D \times M$ $D \approx 10$	$J \times M \times (D+1)$ $10 \leq D \leq 50$	$2 \times J \times M + \Gamma \times (D+4M+1)$ $10 \leq D \leq 50$, $\Gamma \approx 400$
Typical ratio of unknowns / data	0.23	0.78	3.4	0.16
Analysis duration	3 min.	5 min.	30 h.	8 h.

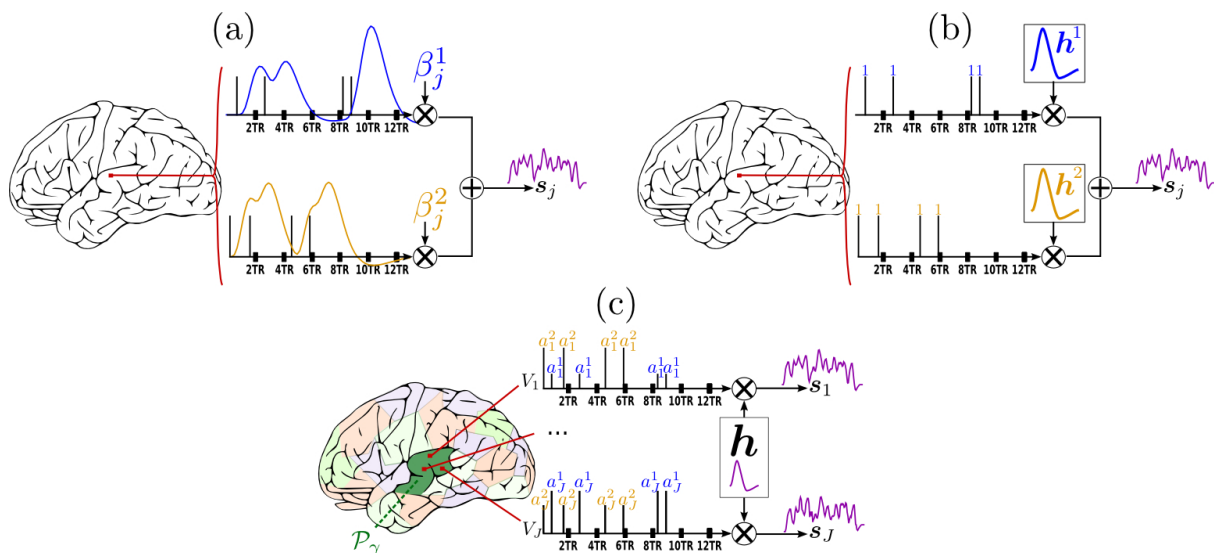


Figure 1. Forward models generating the stimulus-induced components for the methods available in `pyhrf`. In all cases, the scheme involves two experimental conditions colored in blue and yellow with four stimulation events as depicted by vertical bars over the TR-sampled grid. **(a):** General Linear Model (GLM). For a given condition in a given voxel, the stimulus event sequence is convolved with the fixed canonical HRF resulting in a fixed stimulus-induced regressor. This regressor is then multiplied by an unknown effect β_j^m . All the condition-specific regressors are then summed to form the final stimulus-induced signal s_j . **(b):** Finite Impulse Response (FIR) Model. In a given voxel, the stimulus event-sequence is convolved with an unknown FIR vector h^m for each condition to yield a condition-specific component. All components are then summed to form the final stimulus-induced signal s_j . **(c):** Joint Detection-Estimation (JDE). For a given voxel in a given parcel \mathcal{P}_γ , the stimulus sequence gathering all experimental conditions is multiplied by the response levels $\{a_j^m\}$. Then, this spike signal is convolved with an unknown spatially-invariant HRF h to form the stimulus-induced signal s_j .

The generative model, illustrated in Fig. 1(a), reads:

$$\forall j, \quad \mathbf{y}_j = \sum_{m=1}^M \mathbf{X}^m (\beta_j^m \mathbf{h}_c + \beta_j^{l'm} \mathbf{h}'_c + \beta_j^{l'l'm} \mathbf{h}''_c) + \mathbf{P} \ell_j + \mathbf{b}_j, \quad (2)$$

where β_j^m , $\beta_j^{l'm}$, $\beta_j^{l'l'm}$ are the unknown effects associated with the m^{th} stimulus-induced regressors constructed with the fixed known vectors \mathbf{h}_c , \mathbf{h}'_c , \mathbf{h}''_c respectively. Here, the size of \mathbf{h}_c is such as

$D = 25/\text{TR}$ for a duration of 25 sec. To obtain the classical GLM with only the canonical HRF, β'_j and β''_j can be set to zero for all positions. It is worth noting that this formulation of the forward model is equivalent to the classical one where all regressors are gathered in the design matrix (noted $\bar{\mathbf{X}}$) and all corresponding effects gathered in a single vector $\bar{\boldsymbol{\beta}}$. Eq. (2) can be written as:

$$\forall j, \quad \mathbf{y}_j = \bar{\mathbf{X}}\bar{\boldsymbol{\beta}}_j + \mathbf{b}_j, \quad (3)$$

with:

$$\bar{\mathbf{X}} = [\mathbf{X}^1 \mathbf{h}_c | \cdots | \mathbf{X}^m \mathbf{h}_c | \mathbf{X}^1 \mathbf{h}'_c | \cdots | \mathbf{X}^m \mathbf{h}'_c | \mathbf{X}^1 \mathbf{h}''_c | \cdots | \mathbf{X}^m \mathbf{h}''_c | \mathbf{P}]^T,$$

$$\bar{\boldsymbol{\beta}}_j = [\beta_j^1 | \cdots | \beta_j^m | \beta_j'^1 | \cdots | \beta_j'^m | \beta_j''^1 | \cdots | \beta_j''^m | \boldsymbol{\ell}_j]^T.$$

189 The hemodynamics fluctuations caught by such model are limited to ~ 1 second around the peak of the
 190 canonical HRF which is at 5 sec, see **Calhoun et al.** (2004). This model is massively univariate since
 191 every position j is analyzed independently, i.e., no correlation between neighboring signals is considered.
 192 Such model works well on spatially smoothed data to counter-balance the low signal-to-noise ratio, at
 193 the expense of blurred activation clusters. In the `nipy` implementation of the GLM, the fitting process
 194 can be performed using ordinary least square in the case of white Gaussian noise or using Kalman filtering
 195 in the case of an $AR(1)$ Gaussian noise process.

2.2.2 FIR GLM and Regularized FIR

The generative BOLD signal modeling in the FIR context encodes all HRF coefficients as unknown variables:

$$\forall j, \quad \mathbf{y}_j = \sum_{m=1}^M \mathbf{X}^m \mathbf{h}_j^m + \mathbf{P}\boldsymbol{\ell}_j + \mathbf{b}_j \quad (4)$$

196 Here, vector $\mathbf{h}_j^m = (h_{j,d\Delta t}^m)_{d=0,\dots,D}^t$ represents the unknown HRF time course in voxel j which is
 197 associated with the m^{th} experimental condition and sampled every Δt . For the FIR GLM, the size of
 198 \mathbf{h} is such as $D = 25/\text{TR}$ for a duration of 25 sec. Over-sampling could be performed here but is not
 199 advisable in terms of estimability since some FIR coefficients may be poorly or even not associated with
 200 paradigm covariates in matrix \mathbf{X}^m , depending on the paradigm jittering. In its un-regularized version, the
 201 FIR model can be expressed in the GLM framework and hence its implementation in `pyhrf` relies on
 202 `nipy`.

203 In the case of the Regularized FIR (**Ciuciu et al.** (2003)), the problem is placed in the Bayesian
 204 formalism in order to inject regularity on the recovered HRF coefficients \mathbf{h}_j . More specifically, $\mathbf{h}_j^m \sim$
 205 $\mathcal{N}(\mathbf{0}, v_{\mathbf{h}_j^m} \mathbf{R})$ with $\mathbf{R} = (\mathbf{D}_2^t \mathbf{D}_2)^{-1}$ where \mathbf{D}_2 is the second-order finite difference matrix enforcing
 206 local smoothness by penalizing abrupt changes quadratically and $v_{\mathbf{h}_j^m}$ is the unknown HRF prior variance
 207 which is jointly estimated. The size of \mathbf{h} is typically $D = 25/(\text{TR}/4)$ for a duration of 25 sec. and an over-
 208 sampling factor of 4. The fitting process is performed by an Expectation-Maximization (EM) algorithm
 209 to evaluate maximum a posteriori (MAP) voxelwise HRF estimators. For computational and inference
 210 details about this model, see **Ciuciu et al.** (2003).

2.2.3 Joint Detection-Estimation

212 The functional mask of a given subject's brain is a priori divided in Γ functionally homogeneous *parcels*
 213 using methods described in subsection 2.3.2. In each parcel \mathcal{P}_γ , the shape of the HRF \mathbf{h}_γ is assumed

214 constant and the parcel-specific generative model reads:

$$\forall j \in \mathcal{P}_\gamma, \quad \mathbf{y}_j = \sum_{m=1}^M a_j^m \mathbf{X}^m \mathbf{h}_\gamma + \mathbf{P} \ell_j + \mathbf{b}_j. \quad (5)$$

215 where \mathbf{y}_j , \mathbf{X}^m , \mathbf{P} , ℓ_j and \mathbf{b}_j match the variables introduced in subsection 2.2.1. As for RFIR, the size
 216 of \mathbf{h} is typically $D = 25/(\text{TR}/4)$ for a duration of 25 sec. and an over-sampling factor of 4. Here a_j^m
 217 stands for the Neural Response Level (NRL) in voxel j for condition m . As shown in Fig. 1(c) which
 218 illustrates this forward model, the variable a_j^m encodes fluctuations that occur *before* the application of
 219 the hemodynamic filter. Therefore, they are assimilated to neural effects, hence termed “Neural Response
 220 Levels”. However, this term, which is historical, might be misleading as it is difficult to disentangle the
 221 contribution of the neural and the vascular components from single BOLD fMRI data. **These terms can
 222 be more simply identified to the voxel- and condition-specific response amplitudes.**

223 In contrast to Eq. (2) for the GLM forward model, the fixed HRF components \mathbf{h}_c and \mathbf{h}'_c are replaced
 224 by an *unknown* parcel-based HRF \mathbf{h}_γ . Similarly, each unknown NRL a_j^m embodies a single magnitude
 225 parameter per regressor whereas the GLM formulation implies that the magnitude is distributed between
 226 weights β_j^m , β_j^m and β_j^m . To summarize, the HRF shape and the BOLD response magnitude are coupled
 227 in the GLM formulation whereas they are decoupled in the JDE formulation.

228 In the Bayesian framework, priors are formulated to (i) enforce temporal smoothness on the HRF shape
 229 **to perform estimation** in the same manner as for RFIR and (ii) account for spatial correlations between
 230 NRLs through spatial mixture models **to perform detection**, as described in **Vincent et al. (2010)**. The
 231 regularization factor that controls the amount of spatial regularization is jointly estimated and optimized
 232 wrt parcel topology so as to perform an adaptive spatial smoothing. If we are not interested in the
 233 estimation of the HRF and the canonical version seems a reliable choice to the experimenter, then the
 234 HRF can be fixed to its canonical version in the JDE framework which hence amounts to a *spatially
 235 adaptive GLM*. **The latter approach enables parcelwise multivariate detection of activations with adaptive
 236 regularization across parcels. As shown in Badillo et al. (2013b), at the group-level, this strategy retrieves
 237 more peaked and less extended activation clusters compared to classical SPM-like analysis.**

238 The inference is performed by a stochastic sampling scheme where posterior mean estimates (or MMSE)
 239 are computed from Markov Chain Monte Carlo samples. The implementation of the main sampling loop
 240 is coded in pure python and some intensive samplers such as the one for the HRF of the NRLs are coded
 241 in C to save computation time. Still, the overall JDE procedure is computationally intensive and a whole
 242 brain analysis takes around 10 hours on a single CPU ($N = 128$, $\Gamma = 400$, $M = 10$). However, since there
 243 are as many *independent* models as parcels, the analysis can be split up into parcel-wise *parallel* analyses.
 244 For specific details about parallel computing, see section 3.3. From a methodological point of view, note
 245 that the efficiency of the inference scheme has been improved by resorting to a variational formulation of
 246 the JDE **Chauri et al. (2013)** which is also available in `pyhrf`.

2.3 PARCELLATION

2.3.1 Spatial parcellation

248 *Random Voronoi diagrams* A Voronoi diagram consists of a spatial partitioning that builds parcels around
 249 predefined control points or seeds. The parcel boundaries are placed so that each point of a given parcel is
 250 closer to the associated parcel seed than any other seed in terms of the Euclidean distance, as illustrated in
 251 Fig. 2(left). Here, the seed positions are randomly chosen and, in the case of a volume data analysis, these
 252 positions are limited to a shrunk functional mask so that no seed is placed at the edge of the brain, avoiding
 253 peripheral parcels that would be too flat. To build a parcellation from such partitioning, i.e., to assign each
 254 cerebral position to a parcel identifier, we do not explicitly require the parcel boundaries. Accordingly,

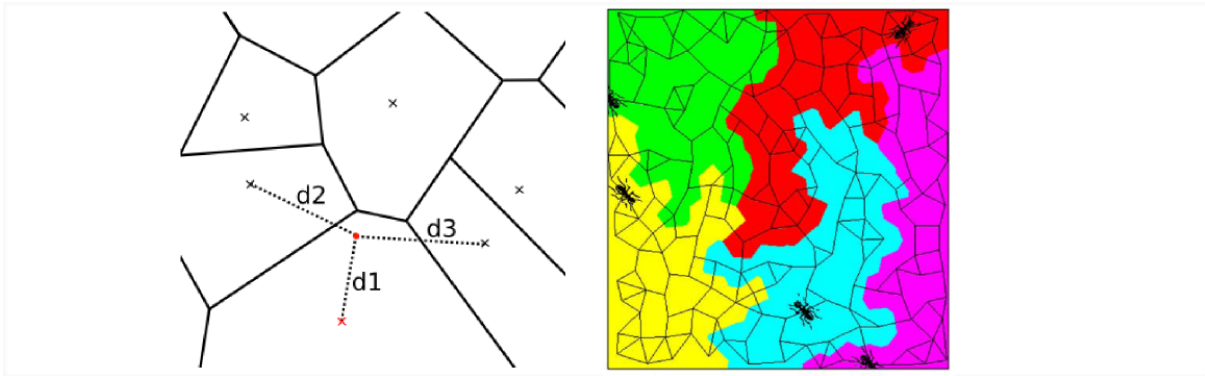


Figure 2. Illustration of spatial parcellation methods in `pyhrf`. **Left:** Voronoi diagram where seeds are represented as crosses. The red point is assigned to the red seed and verifies that its distance to any other seed is larger ($d1 < d2$, $d1 < d3$). **Right:** balanced partitioning performed by patrolling `a(gen)ts`, image extracted from **Elor and Bruckstein (2009)**.

255 there is no need to rely on classical algorithms that precisely compute these boundaries. Instead, a given
 256 position is assigned to the id of the closest seed by resorting to a kd-tree ⁽²⁾.

257 Random Voronoi parcellations are convenient ways to generate samples in the space of sensible
 258 parcellations as they produce convex and compact parcels which are physiologically plausible. They have
 259 been used in **Vincent et al. (2008)** to study the sensitivity of the parcel-based Joint Detection-Estimation
 260 method.

261 *Balanced partitioning* The goal of balanced partitioning is to build parcels of equal sizes. In the case of
 262 a non-regular topology such as the brain, there is no morphological tool to deterministically solve such
 263 a partitioning problem which is known to be NP-complete as mentioned in **Andreev and Racke (2004)**.
 264 Hence, the algorithm implemented in `pyhrf` employs an heuristic and relies on a multi-agent system
 265 that mimics the inflation of balloons in a fixed volume (**Elor and Bruckstein (2009)**), as illustrated in
 266 Fig. 2(right).

267 Balanced partitioning is useful to test the effect of parcel size. In `pyhrf`, balanced partitioning is
 268 implemented in pure python with a position-wise main loop and is hence rather slow: ~ 1 minute to
 269 split 6000 voxels into 20 parcels. However, this performance is sufficient since we only employ balanced
 270 partitioning in the case of small scale testing data sets or when parcels obtained on real data are too big and
 271 they would slow down the overall computation too much, especially in the case of distributed computing.

272 2.3.2 Functional parcellation

273 The main goal of functional parcellation is to provide homogeneous parcels with respect to
 274 hemodynamics. It is mainly motivated by the JDE procedure which assumes that the HRF shape is
 275 constant within one parcel, allowing spatial aggregation within the forward modeling. To provide such
 276 parcellation, results obtained from a GLM analysis, or any given task-specific functional maps are
 277 clustered using different available algorithms: K-means, Ward or spatially-constrained Ward as provided
 278 by `scikit-learn`³. To objectively choose an adequate number of parcels, theoretical information
 279 criteria have been investigated in **Thyreau et al. (2006)**: converging evidence for $\Gamma \approx 400$ at a spatial
 280 resolution of $3 \times 3 \times 3 \text{ mm}^3$ has been shown for a whole brain analysis leading to typical parcel sizes
 281 around a few hundreds voxels ($\approx 2.7 \text{ cm}^3$). As the parcel size is not fixed, some big parcels may arise
 282 from the parcellation process and may slow down the overall parallel processing. To overcome this, the

² implemented in `scipy.spatial.KDTree`

³ `sklearn.cluster.ward`

283 maximum parcel size was controlled by splitting too big parcels (larger than 1000 voxels) according to
 284 the balanced partitioning presented in section 2.3.1, which also guarantees the spatial connexity and thus
 285 properly satisfies the JDE assumptions on the HRF.

286 Such “hard clustering” approach yields sharp parcel boundaries so that smooth transitions between HRF
 287 territories cannot be captured. To avoid wrong boundaries, one can resort to over-segmented parcellations
 288 (high number of parcels) so that transitions may be better captured.

3 PYHRF

289 The installation of `pyhrf` relies on the `setuptools` python package and requires the following
 290 dependencies: `numpy` and `scipy` for core algorithms, `nibabel` for nifti or gifti input/outputs, `nipy` for
 291 the GLM implementation and parcellation tools, `matplotlib` for plots and `PyQT4` for GUIs. Optional
 292 dependencies comprise `joblib`, `scikit-learn` and `soma-workflow`. `pyhrf` is mainly intended
 293 for linux-based distributions as it has especially been developed under Ubuntu. Installation notes and
 294 documentation can be found online at <http://www.pyhrf.org>. Withing the package, the following
 295 data files⁴ are shipped:

- 296 • 2 volumic fMRI data sets (paradigm as CSV files, anatomical and BOLD data files). One serves
 297 quick testing while the other is intended for validation/demonstration purpose, which is used to
 298 generate results in section 4.3,
- 299 • 1 surfacic fMRI data set mainly intended for testing,
- 300 • several simulation resources in the form of png images to provide 2D maps of various activation
 301 labels and HRF territories.

302 The rest of this section is organized as follows. First, the overall workflow of how to use `pyhrf` is
 303 presented, which mainly resorts to command lines and some dedicated GUI tools. Second, to go further
 304 into the package architecture and also to address some features available when scripting, the design of
 305 `pyhrf` is introduced. Third, distributed computation is explained in terms of resource handling. Finally,
 306 the `pyhrf` viewer is presented with a focus on ergonomics.

3.1 WORKFLOW

307 The typical usage of `pyhrf` relies on shell commands which work on XML files. This XML format
 308 was chosen for its hierarchical organization which suits well the nested nature of the algorithm
 309 parametrizations. A dedicated XML editor is provided with a `PyQt4` graphical interface for a quicker
 310 edition and also a better review of the treatment parameters. When such an XML setup file is generated,
 311 it defines a default analysis which involves a small volumic real data set shipped with the package. This
 312 allows for a quick testing of the algorithms and is also used for demonstration purpose. Here is a typical
 313 example of shell commands sequence used to perform a JDE analysis:

```
$ pyhrf_jde_buildcfg -o jde.xml      # generate a default XML file
$ pyhrf_xmledit jde.xml             # set up custom experiment
$ pyhrf_jde_estim -c jde.xml        # run the analysis
$ pyhrf_view *nii                  # view all output nifti files
```

314 The “buildcfg” command offers various options to define setup items from the command line without
 315 having to edit the XML file. For example, the paradigm can be loaded from a CSV or a SPM.mat file.
 316 As for the JDE procedure specifically, the option `--vem` enables the variational EM approach developed
 317 in **Chari et al.** (2013).

⁴ There is no special licence on the shipped data sets.

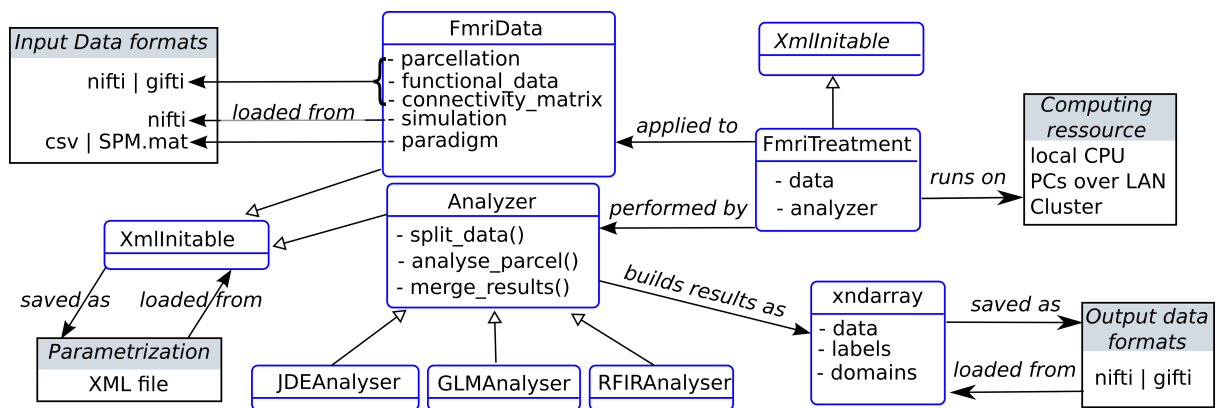


Figure 3. Static organization of the main components in the pyhrf package (not exhaustive). Classes are represented as rounded blue rectangles and external resources (file, computing units) as black rectangles. Note that the XmlInitable class is duplicated for layout convenience. As in UML class diagrams, arrows have the following meaning: → stands for an association, → stands for a generalization.

3.2 DESIGN

318 An overview of the static design of the main package components of the package is shown in Fig. 3. The
 319 class `FmriData` is the within-subject fMRI data representation, irrespective of the spatial support: on the
 320 cortical surface, in the volume, or from a simulation. The common structure to these various data types
 321 comprises spatially flat data (fMRI time series and parcellation) and a connectivity matrix which holds the
 322 data topology. At the centre of the analysis component is the `Analyzer` class that handles parcelwise data
 323 splitting which is done according to the input data parcellation by default, and also takes care of parcel-
 324 specific outputs that are merged at the end of the analysis. This `Analyzer` class is then specialized into
 325 various method-specific analyzers: GLM, RFIR and JDE, FIR being just a specific parametrization of
 326 the GLM. Note that the analyzer component is decoupled from the data component, as classically done
 327 in scientific programming because they do not have the same life-cycles (e.g., the same model can be
 328 applied to various data objects). The `FmriTreatment` packs the data and analysis definitions together
 329 and handles distributed computation across parcels.

330 In the following sub-sections, two specific components are further explained: XML parametrization
 331 through the `XmlInitable` class, and the handling of arrays with axis semantics through the `xndarray`
 332 class.

333 **3.2.1 XML parametrization** The XML format was chosen for its hierarchical organization which suits
 334 the nested nature of the algorithm parametrizations. Indeed, for a JDE analysis, here is an example of
 335 such different levels: `treatment` → `analyzer` → `sampler` → `hrf_sampler`. At a given level,
 336 different classes may be used as there exist, for example, different `sampler` types depending on the type
 337 of prior expressed in the JDE model, so that we require a seamless parametrization process that avoids
 338 rewriting code for the building of parameter files each time a new model is tested. To do so, any object
 339 whose initialization has to be exposed in the XML configuration file inherits the `XmlInitable` class.
 340 This system is not a serialization process as the whole python object is not dumped in the XML. Only the
 341 parameters provided to the `__init__` function are stored. In terms of object life cycle, this process handles
 342 object creation but is not able to track any subsequent modification. Fig. 4 shows a python code sample
 343 that illustrates how the XML file is generated from this nested configuration situation. The resulting XML
 344 file as viewed by the command `pyhrf_xmledit` is also displayed.

345 **3.2.2 The `xndarray` class: data array with axis semantics** The development of semantics-driven
 346 operations on data arrays were motivated by the parcel-driven nature of the analysis workflow which

```

from pyhrf.xmlio import XmlInitable, to_xml
import numpy as np

class FmriTreatment(XmlInitable):
    def __init__(self, input_data=None,
                 analysis_parameters=None):
        XmlInitable.__init__(self)

data = { 'bold_file' : './my_bold.nii',
         'paradigm' : np.array([0,2.3,6.]) }

analysis = { 'model' : 'JDE_MCMC',
             'mcmc_sampling' : {
                 'HRF' : { 'duration' : 25,
                          'type' : 'canonical' }}}

treatment_xml = to_xml(FmriTreatment(data, analysis))
f = open('./test.xml', 'w')
f.write(treatment_xml)
f.close()

```

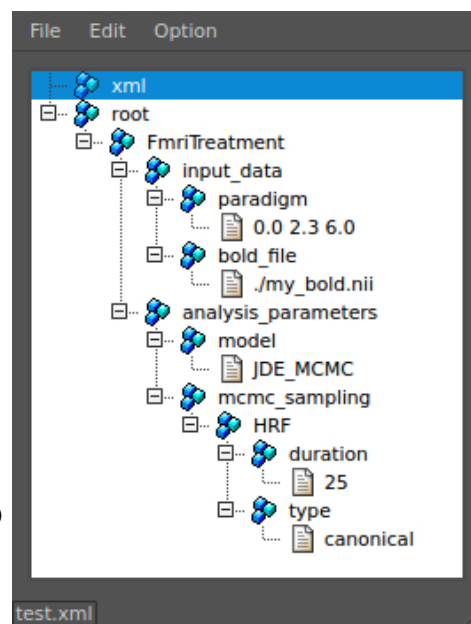


Figure 4. Handling of XML parametrization. The left part shows a code snippet that defines a dummy yet typical fMRI treatment structure with nested components. The init process of the resulting top-level object is then saved in an XML file. The right part is a snapshot of the `pyhrf.xmlioedit` main window where the XML file generated by the code snippet is browsed.

347 implied that parcel-specific results have to be merged in a transparent fashion, whatever their shape.
 348 Indeed, as `pyhrf` is the repository of all the methodological tools developed within the JDE framework,
 349 the number and the form of outputs is highly changing during the development and testing process.
 350 This involves producing convergence tracking, intermediate quantities in addition to the final results
 351 of interest. To avoid writing specific saving procedure for such versatile and numerous outputs, the
 352 information about the interpretation of the data axes has to be explicit. The class `xndarray` handles
 353 any required reorientation prior to saving data arrays into nifti or gifti files. In the volumic data case, the
 354 reorientation follows the `nibabel` convention that is sagittal, coronal, axial and time. To store the extra
 355 axis information along with the data, a dedicated nifti-extension is also written in the volumic data case
 356 or add a “`pyhrf_xndarray_data`” field in the gifti meta data dictionary in the surfacic data case.

357 Moreover, outputs are primarily generated at the parcel-level so that they are in a flat shape, i.e., the
 358 position axis represent indexes of positions in the spatial domain. To form the final whole brain outputs,
 359 the parcel-specific outputs have to be merged together and the position axis, if present, has to be mapped
 360 into the final spatial domain. Table 2 shows two examples of parcel-specific outputs that are merged to
 361 form whole brain data either by spatial mapping or by parcel stacking. To handle these two merging
 362 operations, `stack` and `merge` functions are provided. The reverse process is also available via the
 363 method `explode` which allows an array to be splitted according to a mask composed of integers, ie
 364 a parcellation. It returns the dict of 'flat' parcel-specific data arrays associated with each integer label
 365 present in the mask.

366 In terms of data life cycle, `xndarray` objects are used to prepare data before analysis and to
 367 pack results after analysis. During the analysis process, it is more convenient to work with `numpy`
 368 arrays directly. The following code snippet illustrates the usage of `xndarray` objects: functional and
 369 parcellation data are loaded, within-parcel means are computed and the results is saved to nifti:

```

from pyhrf.ndarray import xndarray, merge
# Data loading
func_data = xndarray.load('./bold.nii')

```

```

parcellation = xndarray.load('./parcellation.nii')
# Split functional data into parcel-specific data
parcel_fdata = func_data.explode(parcellation)
# Fill parcel-specific data with spatial means
parcel_means = dict( (parcel_id, d.copy().fill(d.mean('position'))))
                    for parcel_id,d in parcel_fdata.items() )
# Merge parcel-specific means (map 'position' axis onto spatial axes)
parcel_means = merge(parcel_means, parcellation, axis='position')
# Save output
parcel_means.save('./bold_parcel_means.nii')

```

3.3 DISTRIBUTED COMPUTING

370 PyHRF provides parallel processing features by exploiting local resources (multiple processors on a single
 371 workstation) as well as remote parallel processing units such as a local grid network or a cluster. A whole
 372 brain JDE analysis then boils down from 10 hours to 15 minutes in parallel (on a 100-cores cluster). More
 373 precisely the available computing resources are handled as follows:

- 374 • **local multiple-cores CPUs:** through the use of `joblib` parallel features. The latter works by
 375 spawning python sub-processes that are then run on the different processing units by the operating
 376 system. The number of used CPUs can be setup by the user.
- 377 • **machines over a local area network:** through in-house code that relies on `paramiko` and
 378 hence uses `ssh` connections to distribute jobs on the LAN. A basic scheduler is implemented in
 379 `pyhrf.grid` that can also report faulty remote runs.
- 380 • **multiple-cores cluster:** through `soma-workflow`⁵ developed by **Laguitton et al.** (2011), which
 381 relies on `paramiko`⁶ on the client side and on `DRMAA`⁷ on the server side.

⁵ <http://brainvisa.info/soma-workflow/>

⁶ <http://www.lag.net/paramiko/>

⁷ <http://www.drmaa.org/>

Table 2. Examples of merging operations performed on multiple parcel-specific data arrays, for some JDE outputs: parcel-specific HRFs and condition- and voxel-specific activation labels. If the `xndarray` object contains the "position" axis, as for the "labels" object, then all parcel-specific results are merged into the same target volume and we depict the spatial mapping operation as "→" to map the "position" axis in to the spatial axes "axial", "coronal" and "sagittal". For other axes aside from "position", no merging operation is performed ("=" symbol). If the `xndarray` object does not contain the "position" axis, as for the HRF object, then all parcel-specific results are stacked and a new "parcel" axis is created ("∪" symbol).

	Parcel-specific flat data		Merging operation	Whole brain data	
	axis label	axis domain		axis label	axis domain
HRF	time	[0, ..., hrf_duration]	=		same
			∪	parcel	[0, ..., parcel_max]
labels	class	['activ', 'non_activ']	=		same
	condition	['audio', 'video']	=		same
	position	[0, ..., pos_max]	→	axial	[0, ..., axial_max]
				coronal	[0, ..., coronal_max]
				sagittal	[0, ..., sagittal_max]

382 The distribution problem addressed here is a so-called embarrassingly parallel problem where the same
383 treatment has to be repeated on several parcel-specific pieces of data. There is no shared memory
384 management between distributed processes here.

385 To optimize the distribution process, the order in which the parcel-specific treatments are pushed in the
386 process queue is done by pushing the biggest parcels first. In the same optimization purpose, a safeguard is
387 imposed on the maximum parcel size (more than 7 cm³ in the volume or 11 cm² on the surface). If a parcel
388 exceeds this limit, it is divided up according to the balanced partitioning presented in sub-section 2.3.1.

3.4 VIEWER

389 `pyhrf_view` is a dedicated viewer built on `PyQt4` which embeds a `matplotlib` view. The purpose
390 of `pyhrf_view` is to provide convenient browsing into volumic data⁸. However, it does not provide
391 advanced overlaying features such as the display of functional over anatomical data. Instead, to plot the
392 final “publication-ready” maps after having selected the results of interest with `pyhrf_view`, one can resort
393 to the command `pyhrf_plot_slice` to directly generate a slice image of functional rendering along with
394 anatomical overlay. One can also use a third party viewer such as `Anatomist`⁹, `FSL_view`¹⁰ or `xjview`¹¹.

395 `pyhrf_view` offers n-dimensional browsing while most viewers in neuro-imaging software handle up
396 to 4D volumes. In fact, there is a limit to the number of dimensions inherent to the nifti format which
397 permits 7 axes at maximum. The viewer is composed of two main components (see 5:

- 398 • a main window handling object and slice selection,
- 399 • plot windows which display the selected slice as curve or image.

400 The slice selection tools provides sliders to browse through axes domain values and display related
401 information: axis name, current selected domain values and projection states. There can be up to two
402 projected axes (2D), i.e., axes which will mapped to the actual plot axes. When multiple objects are loaded,
403 slicers are synchronized to plotting views so that click events yield slider updates. This behavior can be
404 modified in two ways. First, the reception combo box toggles whether the slider receives changes from
405 other sliders. This is useful when one wants to prevent a given view from being updated by synchronization
406 events (with reception off), e.g., when a reference slice should be compared to other slices. Second, the
407 emission combo box toggles the spreading of slider changes to all other slicers. This is typically used to
408 control a given axis across all displayed objects with a single slider (with emission on).

4 RESULTS

4.1 EXPERIMENTAL PARADIGM

409 In all presented results, whether they focus on artificial or real data sets, we resorted to the same
410 experimental paradigm. The latter is a multi-functional cognitive localizer paradigm designed in **Pinel**
411 **et al.** (2007). This paradigm enables to map cognitive brain functions such as reading, language
412 comprehension and mental calculations as well as primary sensory-motor functions. It consists of a *fast*
413 *event-related* design (sixty stimuli) comprising the following experimental conditions: auditory and visual
414 sentences, auditory and visual calculations, left/right auditory and visual clicks, horizontal and vertical
415 checkerboards. The average ISI is 3.75 sec. including all experimental conditions. Such a paradigm
416 is well-suited for simultaneous detection and estimation, in contrast to slow event-related and block
417 paradigms which are considered optimal only for estimation or detection, respectively (**Liu et al.** (2001)).

⁸ Surface rendering is not available. `Anatomist` is recommended for such usage

⁹ <http://brainvisa.info>

¹⁰ <http://fsl.fmrib.ox.ac.uk/fsl/fslview/>

¹¹ <http://www.alivelearn.net/xjview8/>

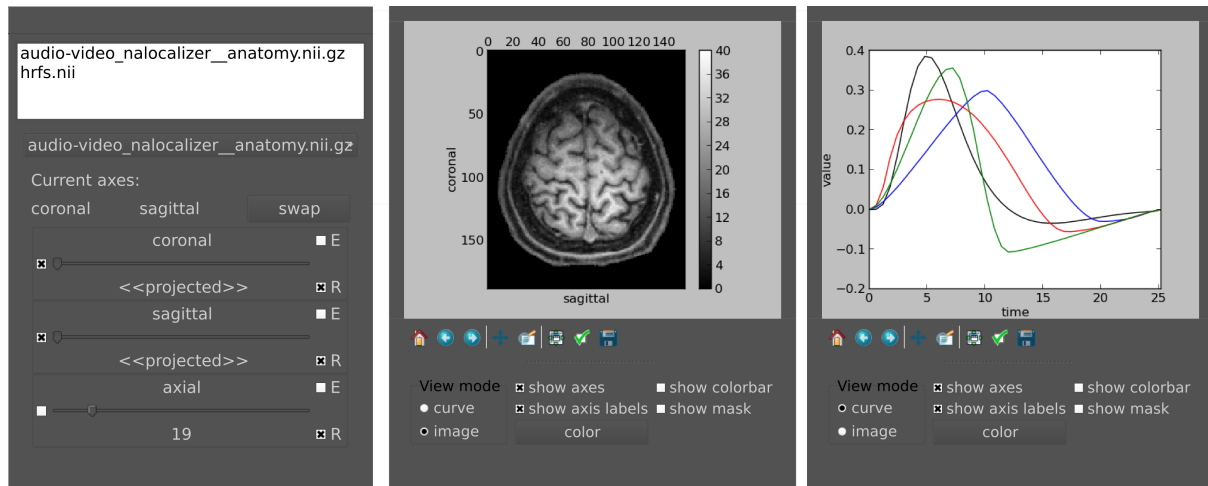


Figure 5. Main widget components of `pyhrf-view` to browse and view n-dimensional data. **Left:** the list widget on top displays the currently loaded objects. The slicer panel at the bottom allows: projection of axes (combo boxes on the left), domain value slicing (sliders in the middle) and definition of view synchronization (combo boxes on the right). For a given axis slicer, the two combo boxes defining synchronization are: (E) toggle emission of slice change to other slicers, (R) toggle reception from other slicers or from click events on plots. **Middle:** plot window for the current selected slice. The top part displays the actual plot as produced by `matplotlib.pyplot`. The bottom part offers changing the view mode (either curve, image, or histogram), and toggling display of axes, colorbar and mask. The color button pops up a gradient map selector if in image mode or a color picker if in curve mode. **Right:** other plot window to illustrate curve display.

4.2 ARTIFICIAL DATA GENERATOR

418 Simulations in `pyhrf` mainly consists of building a script that defines a simple pipeline organization.
 419 Indeed, the process of generating fMRI data involves many versatile simulation bricks with various
 420 dependencies between them as shown in Table 3 which presents the generation processes available in
 421 `pyhrf`. Writing a simulation script as a sequence of functions makes things difficult to read and to reuse.
 422 Instead, all simulation bricks are gathered inside a python dictionary that maps a simulation label to its
 423 corresponding value. This value can be directly defined as a python object or as a function which can
 424 depend on other simulation items and which is called when the simulation pipeline is evaluated. The
 425 pipeline structure arises from the link between simulation labels and function arguments. In practice, an
 426 example of such simulation script is as follows:

```
import numpy as np
from pyhrf.ndarray import xndarray
from pyhrf.tools import Pipeline

# Functions used to generate items in the simulation Pipeline
def generate_rls(spatial_shape, mean_rls, var_rls):
    rls = np.random.randn(*spatial_shape) * var_rls**.5 + mean_rls
    return xndarray(rls, ['axial', 'sagittal', 'coronal'])

def generate_noise(stim_induced_signal, noise_var):
    noise = np.random.randn(*stim_induced_signal.data.shape) * noise_var**.5
    return xndarray(xndarray_like(stim_induced_signal, data=noise))

def create_stim_induced_signal(rastered_paradigm, hrf, response_levels):
    signal = np.convolve(rastered_paradigm, hrf)[np.newaxis,:] * \
        response_levels.data[:, :, :, np.newaxis]
    return xndarray(signal, response_levels.axes_names + ['time'])
```



```

def create_bold(stim_induced_signal, noise):
    return stim_induced_signal + noise

# Definition of the simulation pipeline
simulation_steps = {
    'spatial_shape' : (10,11,12), 'mean_rls' : 3., 'var_rls' : 0.5,
    'response_levels' : generate_rls,
    'rastered_paradigm' : np.array([0,0,1,0,0,0,1,0,0,0,1]),
    'hrf' : np.array([0,.5,1,0.5,0.,0]),
    'noise_var' : 1.,
    'noise' : generate_noise,
    'stim_induced_signal' : create_stim_induced_signal,
    'bold' : create_bold,
}

simulation = Pipeline(simulation_steps)

# Computation of all quantities in the pipeline and data saving
simulation.resolve()
simulation_items = simulation.get_values()
simulation_items['response_levels'].save('./response_levels.nii')
simulation_items['stim_induced_signal'].save('./stim_induced_signal.nii')
simulation_items['bold'].save('./bold.nii')

```

427 The artificial data experiment presented here comprises the generation of BOLD time series within the
428 volume and then projected onto the cortical surface. To do so, shipped data defines a volume of 4 HRF
429 territories, as well as the grey/white matter segmentation obtained from real data in the occipital region.
430 Within the grey matter mask, activation labels are generated and conditionally to them, response levels
431 are simulated according to a bi-Gaussian mixture. For the sake of simplicity, a version of the localizer
432 paradigm presented in the previous section is merged over the auditory and visual modalities so as to
433 obtain only two conditions. In all HRF territories this paradigm is then convolved with HRF generated by
434 Bezier curves that enable the control of the time-to-peak and time-to-undershoot. Finally, nuisance signals
435 are added (Gaussian noise and polynomial drift) to obtain the volume of artificial BOLD data. To generate
436 surfacic data, data are projected on a cortical fold that is also shipped in the package and we resorted to an
437 external projection tool, developed in **Operto et al. (2006)** but others are available such as *Freesurfer*.
438 Fig. 6 presents the results obtained on artificial data using the JDE procedure. HRF estimates recover their
439 respective ground truth profiles with a slightly more deformed curve obtained on the cortical surface for
440 the bottom right (green) HRF territory, compared with the volumic data case. Detection results (response
441 levels maps in Fig. 6) also shows the correct recovery of the simulated ground-truth, in the volume and on
442 the cortical surface.

Table 3. Different types of simulation bricks available in *pyhrf*. The “localizer” paradigm is described in **Pinel et al. (2007)**. Hand-drawn maps for activation labels are in the form of *png* images. Gaussian smooth generation of HRFs stands for the regularized prior used in the JDE model.

Simulation item	available generation process
Experimental paradigm	localizer, random event-related
Activation labels	hand-drawn 2D maps, 3D Potts realizations
Response levels	bi / tri mixture of Gaussian or Gamma components
Hemodynamic response function	canonical, Bezier curve, Gaussian smooth
Low frequency drift	polynomial, cosine
Noise	white, auto-regressive of order p

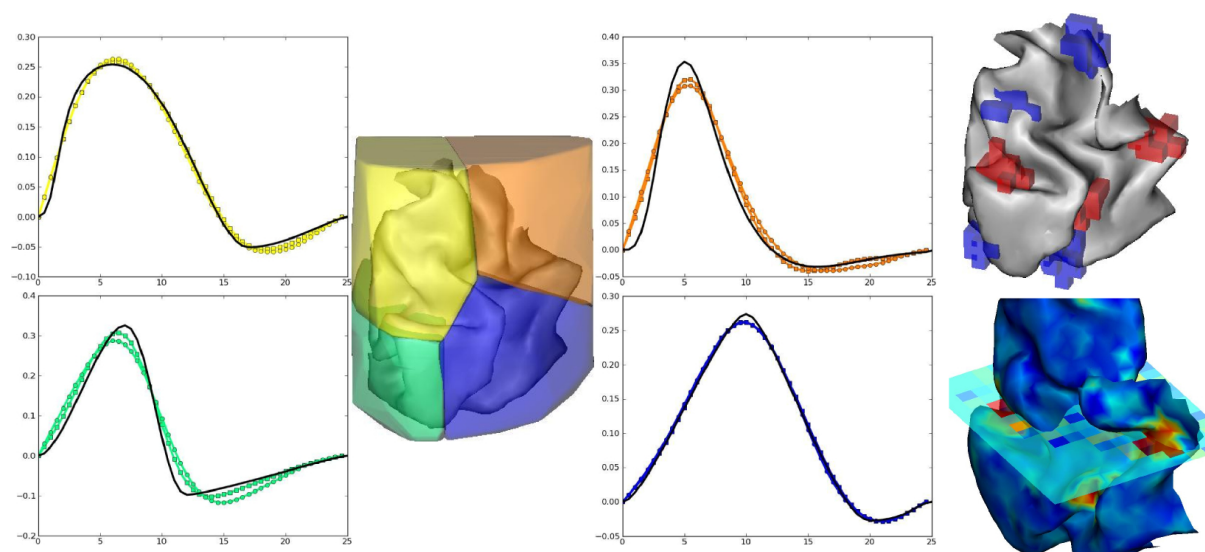


Figure 6. Results on volumic and surfacic artificial data. **Left part:** HRF estimates obtained by JDE on the 4 artificial parcels. Ground truth HRFs are depicted in black line while colored HRF are HRF estimates that match the color of the parcels. **Right part, top:** labels simulated in the cortical fold for two conditions (in blue and red). **Right part, bottom:** response levels estimates obtained by JDE on the cortical surface and in a selected slice of the volume. 3D renderings were produced with *anatomist*.

4.3 WITHIN-SUBJECT METHOD COMPARISON

443 The analyzed real data set, which is shipped with *pyhrf*, was a subset of an fMRI acquisition performed
 444 on a single healthy subject with a 3-Tesla Tim Trio Siemens scanner using an EPI sequence. The following
 445 settings were used for this acquisition: the fMRI session consisted of $N = 128$ scans, each of them being
 446 acquired using $TR = 2400$ ms, $TE = 30$ ms, slice thickness: 3 mm, transversal orientation, $FOV =$
 447 192 mm^2 and spatial in-plane resolution was set to $2 \times 2 \text{ mm}^2$. Data was collected using a 32 channel
 448 head coil to enable parallel imaging during the EPI ($R=2$) acquisition. Parallel SENSE imaging was used
 449 to keep a reasonable Time of Repetition (TR) value in the context of high spatial resolution. In order to
 450 reduce disk usage and to focus only on areas of the brain which are expected to elicit activity in response
 451 to the paradigm, functional data was restricted to selected regions of interest that comprise occipital,
 452 temporal, parietal and motor regions. To improve interpretation and data plot rendering, an anatomical
 453 image is also shipped, with an in-plane resolution of $1 \times 1 \text{ mm}^2$ and slice thickness of 1.1 mm.

454 This fMRI data set was analyzed using GLM with a canonical HRF, FIR, RFIR and JDE¹². For JDE, the
 455 functional parcellation was built according to the method described in section 2.3.2. Fig. 7(a-b) depicts
 456 detection results for the auditory effect, obtained by GLM with canonical HRF (see Fig. 7(a)) and JDE
 457 (see Fig. 7(b)). Both methods highlight the same activation localization, with a slightly stronger sensitivity
 458 for JDE. Fig. 7(c) shows HRF estimation results as obtained by FIR, RFIR and JDE at the same local
 459 maximum on the left temporal region. Note that the HRF estimate provided by the JDE procedure is
 460 regional. The HRF profile delivered by FIR appears noisier than the JDE and RFIR counterparts. Also the
 461 temporal resolution of FIR is limited to the TR of input data. In contrast, RFIR and JDE offer an enhanced
 462 temporal resolution of 0.6 sec. In terms of timing, the FIR and JDE methods yield a peak at 5 seconds
 463 which is compatible with the canonical HRF that has been fitted on temporal auditory regions (Boynton
 464 et al. (1996)). Accordingly, the HRF estimates obtained by RFIR seems over-smoothed. Overall, JDE
 465 enables reliable activation maps and HRF profiles which can roughly be obtained by separate GLM and
 466 FIR analyses. Fig. 7(d-e) shows results on effect maps for the computation effect, obtained by GLM with
 467 canonical HRF (see Fig. 7(d)) and JDE (see Fig. 7(e)). JDE results have a higher sensitivity which can be

¹² analysis scripts are available at <http://github.com/pyhrf/pyhrf/tree/master/script/frontiersBIM14/>

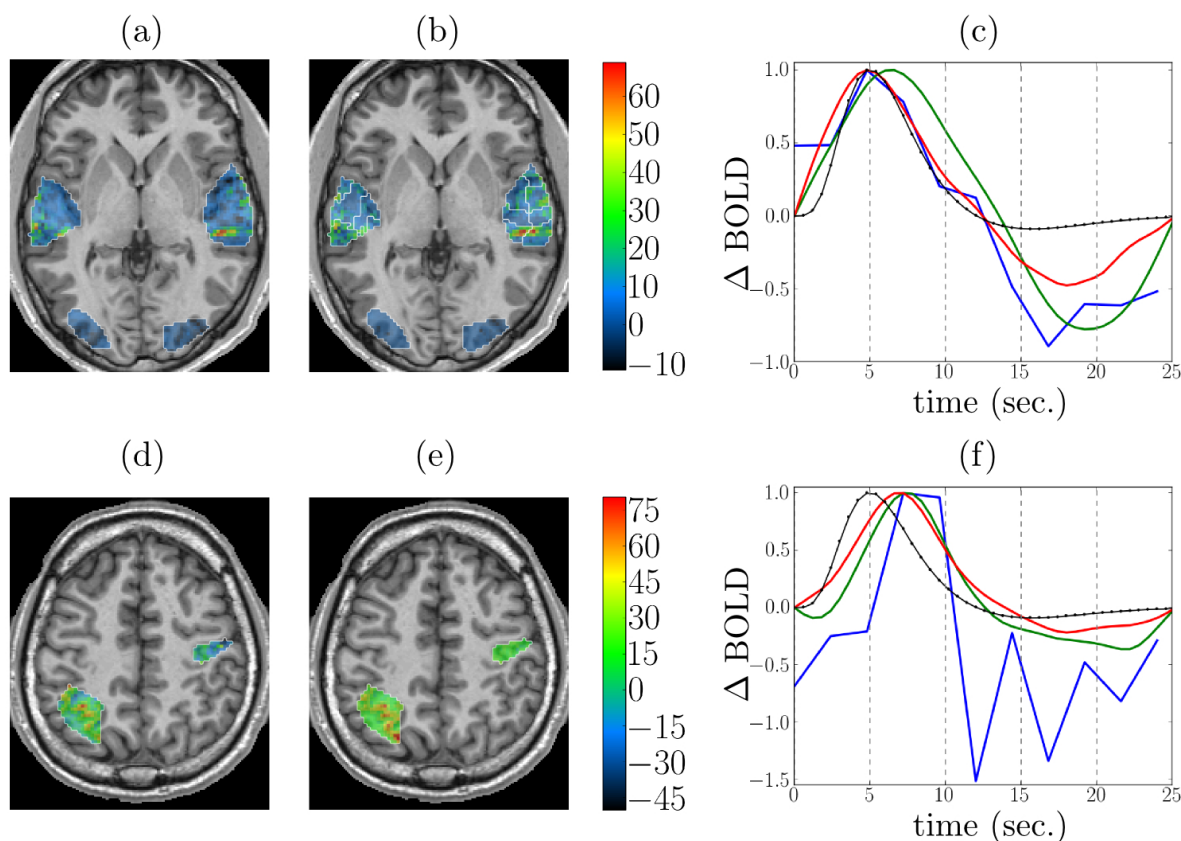


Figure 7. Detection and estimation results on the shipped real data set. Top and bottom rows: auditory and computation experimental conditions, respectively. Columns from left to right: response level maps, for (a,d) GLM with canonical HRF, (b,e) JDE, superimposed with the functional parcellation (white borders). Neurological convention: left is left. (c): Estimation results for GLM FIR (blue), RFIR (green) and JDE (red). The canonical HRF is shown in black.

468 explained by an estimated HRF that differs from the canonical version (see Fig. 7(f)). More specifically
 469 on the HRF estimation results shown in Fig. 7(f), we can draw the same comments as for the auditory
 470 results. However, the FIR HRF profile is here more chaotic and its peak is less easy to identify as the
 471 curve shows a plateau between 7 and 10 sec.

4.4 GROUP-LEVEL HEMODYNAMICS

472 Using `pyhrf`, the hemodynamic variability was also studied on a group of 15 healthy volunteers (average:
 473 23.2 years, std: 2 years). The experimental paradigm is described in Section 4.1 and the fMRI acquisition
 474 parameters are similar to those previously mentioned in subsection 4.3. The results presented hereafter
 475 have been published in **Badillo et al.** (2013b). In this work, hemodynamic variability was investigated
 476 in four regions of interest, located in the left parietal cortex (*P*), bilateral temporal (*T*) and occipital (*O*)
 477 lobes and in the right motor cortex (*M*), as shown in Fig. 8. These regions were defined after conducting
 478 a random-effect analysis to detect activation clusters showing a significant group-level effect. More
 479 precisely, we defined four contrasts of interest targeting brain activity in sensory and cognitive regions:
 480 a *Auditory vs. Visual* contrast for which we expect evoked activity in temporal regions in response, a
 481 *Visual vs. Auditory* contrast that induces evoked activity in the occipital cortex, a *Left vs. Right click*
 482 contrast for which we expect evoked activity in the right contralateral motor cortex, and a *Computation*
 483 *vs. Sentence* contrast which is expected to highlight activity in the frontal and parietal lobes specific to
 484 mental calculations. In terms of detection performance, at the group-level, JDE and GLM are comparable
 485 in primary sensory regions (where the canonical HRF is appropriate). However, in the parietal region

486 involved in higher cognitive processes, the JDE approach yields more sensitive maps. In what follows, we
 487 summarize group-level hemodynamics results obtained in the regions of interest extracted from activated
 488 clusters.

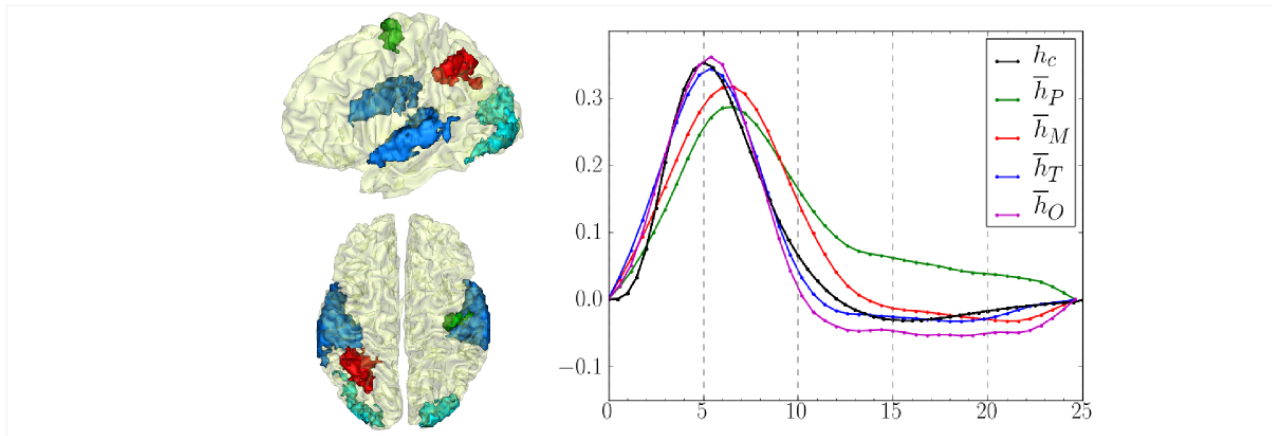


Figure 8. **Left:** Definition of regions of interest to investigate hemodynamics variability from JDE-based group-level analysis. **Top:** Sagittal view. **Bottom:** axial/top view. Left parietal area (P) appears in red, left motor area in the pre-central cortex is shown in green, Bilateral temporal regions along auditory cortices and bilateral occipital regions in the visual cortices are shown in blue and cyan, respectively. **Right:** Group-average HRF estimates for the four regions of interest: \bar{h}_P , \bar{h}_M , \bar{h}_T , \bar{h}_O stand for HRF means in parietal, motor, temporal and occipital regions, respectively. h_c correspond to the canonical HRF.

489 The group-level HRF extraction in each ROI involves the following steps: For each subject, we
 490 identified the parcel containing the mostly activated voxel across stimulus-dependent response levels.
 491 Each individual parcel-based HRF time course is then scaled by the corresponding maximum response
 492 level so as to account for the inter-subject variability of the effect size. Last, each group-level HRF
 493 profile (see Fig. 8) is computed as the average over the 15 subjects in the corresponding ROI.

494 One of the main results concerns the spatial gradient of discrepancy to the canonical HRF shape between
 495 regions. As shown in Fig. 8, the mean HRF time courses retrieved in occipital and temporal regions are the
 496 closest to the canonical shape h_c . In the motor cortex, the HRF deviates a little bit more from the canonical
 497 filter, especially in terms of hemodynamic delay. Finally, the largest discrepancy to the canonical HRF
 498 was found in the parietal region.

5 PERSPECTIVES

5.1 METHODOLOGICAL PERSPECTIVES

499 Methods that derive from former external works (GLM, FIR, RFIR), are mainly used for comparison
 500 purpose and are not the subject of extension. The main methodological developments are currently taking
 501 place in the JDE framework. In fMRI activation protocols, the paradigm usually consists of several runs
 502 repeating similar sequences of stimuli. For an increased stability of HRF estimates that cope with the
 503 between-run variability of the response magnitude, a multi-run extension has been developed in **Badillo**
 504 **et al.** (2013c), consisting of a random-effect heteroscedastic approach. It is particularly useful for pediatric
 505 imaging where runs are short in time. In the same vein of improving within-subject analyses, an approach
 506 to encode the condition-specificity at the parcel level is being developed to enforce non-relevant conditions
 507 to yield null activation, as in **Bakhous et al.** (2013).

508 In terms of computation cost, we mentioned the variational EM version of JDE that has been published
 509 in **Chaari et al.** (2013) and that appeared to be 10 to 30 times faster than its MCMC alternative. We
 510 have also shown that this numerical speed up is not performed at the expense of the result quality in

511 terms of activation maps and HRF estimates. This algorithmic improvement has allowed us to address
512 a more computationally demanding task, namely the joint Parcellation-Detection-Estimation (JPDE)
513 model **Chaari et al.** (2012) that jointly estimate the spatial aggregation support of HRF shapes,
514 also termed parcellation, whereas the current JDE approach relies on a fixed prior decomposition in
515 homogeneous territories. The JPDE validation is still ongoing. In an attempt to solve the same issue, an
516 alternative based on random parcellations and consensus clustering has been recently proposed in **Badillo**
517 **et al.** (2013a).

518 Closely related to the results presented in Section 4.4, a multi-subject extension of the JDE is currently
519 developed to properly account for the between-subject HRF variability and recover a meaningful and
520 potentially less biased group-level HRF profile. Indeed, the group-level results presented so far were
521 computed as a simple mean of multiple within-subject JDE analyses. In presence of outliers, the mean
522 estimator is directly impacted and we thus seek for more robust group-level estimates. This issue can thus
523 be answered either by considering robust group-level averaging techniques (weighted or trimmed least
524 squares, median or Huber M -estimators, ...) or by adding an additional layer in the hierarchical Bayesian
525 modeling. This development trail will bring modification in the core design of `pyhrf` so as to take into
526 account the new “group” data axis.

527 Finally, recent works have opened the path to multi-modality by the processing of Arterial Spin Labeling
528 fMRI data **Vincent et al.** (2013). To analyze such data, physiologically-inspired models are investigated to
529 establish parsimonious and tractable versions of physiological models such as the balloon model described
530 in **Friston and Buechel** (2000); **Buxton et al.** (2004). Hence, for validation purpose, the artificial data
531 generator is also being enriched with the simulation of physiological models.

5.2 PACKAGE PERSPECTIVES

532 In addition to improving the documentation and usability of the current package version, additional
533 developments will be first motivated by the above-mentioned methodological perspectives, namely re-
534 factoring part of the data design to integrate the group-level and multi-session data components. This will
535 mainly involve the modification of the `FmriData` class and the addition of a new `FmriGroupData`
536 class. In this respect, the handling of data input will have to be extended to exploit a hierarchy of
537 subject-specific files.

538 In another respect, we plan on enriching the parcellation component by handling classical atlases such as
539 the Automated Anatomical Labeling (AAL) atlas built by **Tzourio-Mazoyer et al.** (2002), the Brodmann
540 regions (**Brodmann** (1909)) and the Harvard-Oxford atlas (**Desikan et al.** (2006)) available in `FSL`¹³.
541 The goal is to enable the definition of functional parcels that are consistent with previous studies in the
542 literature and also to further investigate the anatomo-functional link by comparing atlas-driven versus
543 data-driven parcellations.

544 In order to offer more user-friendliness, the building of a unified graphical user interface is foreseen,
545 which will gather the XML editor and the viewer while also enabling the selection of the analysis
546 type. We also envisage resorting to wizard interfaces to guide the setup process and deliver contextual
547 documentation. In terms of browsing features, tools to properly explore the surface-based results are
548 currently missing, as we resort to an external tool, `anatomist`. The goal is not to reproduce all the
549 features offered by the latter which enable the output of paper-ready figures through joint volume/surface
550 rendering, data fusion and material handling. We rather think of a simple textured mesh viewer associated
551 with a picking feature in order to synchronize other views. The main usage is to make the selection of
552 a mesh node and the corresponding HRF estimate feasible. For making this surface-based rendering av-
553 available, `mayavi`¹⁴ is an appealing candidate since it has been already intensively used in the python
554 community.

¹³ <http://http://fsl.fmrib.ox.ac.uk>

¹⁴ <http://code.enthought.com/projects/mayavi/>

555 Finally, for the computational efficiency aspect, we plan on incorporating GPU parallel computing
556 features. Indeed, this technology is becoming more and more available and powerful and may also appear
557 cheaper than CPU computing systems (see Owens et al. (2008) for a review). Specifically, the NVIDIA
558 chipsets are easily accessible for general purpose computing through the python package `pyCUDA`¹⁵. A
559 simple test on matrix products with a complexity similar to that of our models showed a gain of one
560 order of magnitude in favor of GPU computations¹⁶ (NVIDIA GeForce 435M graphics card) compared
561 to CPU-based computations (Intel Core M480 @ 2.67GHz) with `numpy`.

6 CONCLUSION

562 The `pyhrf` package provides tools to detect evoked brain activity and estimate the underlying dynamics
563 from fMRI data in the context of event-related designs. Several “reference” methods are available: the
564 GLM, FIR and RFIR approaches, and also more flexible models as provided by the JDE framework. The
565 choice of the analysis tools depends on the experimenter’s question: if simple mappings are required, the
566 GLM is appropriate provided that the HRF is expected to be close to its canonical version, but for finer
567 dynamics estimation, the JDE procedure is more suitable. The design of `pyhrf` allows the handling of
568 volumic and surfacic data formats and also the utilization of several distributed computing resources. The
569 main user interface is done by shell commands where the analysis setup is stored in an XML configuration
570 file. Two graphical components are provided: an XML editor and a n-dimensional volumic data browser.

571 This package provides valuable insights on the dynamics of the cognitive processes that are not available
572 in classical software such as `SPM` or `FSL`. Hence, it offers interesting perspectives to understand the
573 differences in the neuro-vascular coupling of different populations (infants, children, adults, patients,
574 etc.).

REFERENCES

- 575 Andreev, K. and Räcke, H. (2004), Balanced graph partitioning, in Proceedings of the sixteenth annual
576 ACM symposium on Parallelism in algorithms and architectures (ACM, New York, NY, USA), SPAA
577 '04, 120–124
- 578 Aurenhammer, F. and Klein, R. (2000), Voronoi Diagrams. (Elsevier Science Publishing), chapter 5,
579 201–290
- 580 Badillo, S., Varoquaux, G., and Ciuciu, P. (2013a), Hemodynamic estimation based on consensus
581 clustering, in IEEE Pattern Recognition in Neuroimaging (PRNI) (Philadelphia, USA)
- 582 Badillo, S., Vincent, T., and Ciuciu, P. (2013b), Group-level impacts of within- and between-subject
583 hemodynamic variability in fMRI, *Neuroimage*, 82, 433–448
- 584 Badillo, S., Vincent, T., and Ciuciu, P. (2013c), Multi-session extension of the joint-detection framework
585 in fMRI, in 10th International Symposium on Biomedical Imaging (San Francisco, CA), 1504–1507
- 586 Bakhous, C., Forbes, F., Vincent, T., Dojat, M., and Ciuciu, P. (2013), Variational variable selection to
587 assess experimental condition relevance in event-related fMRI, in 10th International Symposium on
588 Biomedical Imaging (San Francisco, CA), 1500–1503
- 589 Boynton, G. M., Engel, S. A., Glover, G. H., and Heeger, D. J. (1996), Linear systems analysis of
590 functional magnetic resonance imaging in human V1, *The Journal of Neuroscience*, 16, 13, 4207–4221
- 591 Brodmann, K. (1909), Vergleichende Lokalisationslehre der Grosshirnrinde (Barth, Leipzig)
- 592 Buxton, R. B., g, K. U., Dubowitz, D. J., and Liu, T. T. (2004), Modeling the hemodynamic response to
593 brain activation, *Neuroimage*, 23, Supplement 1, S220–S233

¹⁵ <http://developer.nvidia.com/pycuda>

¹⁶ benchmark available at <http://wiki.tiker.net/PyCuda/examples/DemoMetaMatrixmulCheetah>

- 594 Calhoun, V. D., Stevens, M. C., Pearlson, G. D., and Kiehl, K. A. (2004), fMRI analysis with the general
595 linear model: removal of latency-induced amplitude bias by incorporation of hemodynamic derivative
596 terms, *NeuroImage*, 22, 1, 252 – 257
- 597 Chaari, L., Forbes, F., Vincent, T., and Ciuciu, P. (2012), Adaptive hemodynamic-informed parcellation of
598 fMRI data in a variational joint detection estimation framework, in 15th Proceedings MICCAI (Springer
599 Verlag, Nice, France), LNCS 7512, (Part III), 180–188
- 600 Chaari, L., Vincent, T., Forbes, F., Dojat, M., and Ciuciu, P. (2013), Fast joint detection-estimation
601 of evoked brain activity in event-related fMRI using a variational approach, *IEEE Transactions on*
602 *Medical Imaging*, 32, 5, 821–837
- 603 Ciuciu, P., Poline, J.-B., Marrelec, G., Idier, J., Pallier, C., and Benali, H. (2003), Unsupervised robust
604 non-parametric estimation of the hemodynamic response function for any fMRI experiment, *IEEE*
605 *Transactions on Medical Imaging*, 22, 10, 1235–1251
- 606 Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., et al. (2006), An
607 automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based
608 regions of interest., *Neuroimage*, 31, 3, 968–980
- 609 Duyn, J. and Koretsky, A. (2011), Novel frontiers in ultra-structural and molecular MRI of the brain, *Curr.*
610 *Opin. Neurol.*, 24, 4, 386–393
- 611 Elor, Y. and Bruckstein, A. M. (2009), Multi-a(ge)nt graph patrolling and partitioning, in Proceedings of
612 the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent
613 Technology - Volume 02 (IEEE Computer Society, Washington, DC, USA), WI-IAT '09, 52–57
- 614 Friston, K. (1998), Imaging neuroscience: Principles or maps?, *Proceedings of the National Academy of*
615 *Sciences of the United States of America*, 95, 796–802
- 616 Friston, K. and Buechel, C. (2000), Attentional modulation of effective connectivity from V2 to V5/MT
617 in humans, *Proceedings of the National Academy of Sciences of the United States of America*, 97, 13,
618 7591–7596
- 619 Friston, K., Holmes, A. P., Poline, J.-B., Grasby, P., Williams, S., Frackowiak, R., et al. (1995), Analysis
620 of fMRI time-series revisited, *Neuroimage*, 2, 45–53
- 621 Henson, R., Andersson, J., and Friston, K. (2000), Multivariate SPM application to basis function
622 characterisations of event-related fMRI responses, volume 11, 468
- 623 Koslow, S. H. and Huerta, M. F., eds. (2013), *Neuroinformatics: An Overview of the Human Brain Project*
624 (Psychology Press)
- 625 Laguitton, S., Rivire, D., Vincent, T., Fischer, C., Geffroy, D., Souedet, N., et al. (2011), Soma-
626 workflow: a unified and simple interface to parallel computing resources, in MICCAI Workshop on
627 High Performance and Distributed Computing for Medical Imaging (Toronto)
- 628 Liu, T., Frank, L., Wong, E. C., and Buxton, R. B. (2001), Detection power, estimation efficiency, and
629 predictability in event-related fMRI, *Neuroimage*, 13, 4, 759–773
- 630 Makni, S., Ciuciu, P., Idier, J., and Poline, J.-B. (2005), Joint detection-estimation of brain activity in
631 functional MRI: a multichannel deconvolution solution, *IEEE Transactions on Signal Processing*, 53,
632 9, 3488–3502
- 633 Makni, S., Idier, J., Vincent, T., Thirion, B., Dehaene-Lambertz, G., and Ciuciu, P. (2008), A fully
634 Bayesian approach to the parcel-based detection-estimation of brain activity in fMRI, *Neuroimage*, 41,
635 3, 941–969
- 636 Marrelec, G., Benali, H., Ciuciu, P., Plgrini-Issac, M., and Poline, J.-B. (2003), Robust Bayesian
637 estimation of the hemodynamic response function in event-related BOLD MRI using basic
638 physiological information, *Human Brain Mapping*, 19, 1, 1–17
- 639 Ogawa, S., Lee, T., Kay, A., and Tank, D. (1990), Brain magnetic resonance imaging with contrast
640 dependent on blood oxygenation, *Proceedings of the National Academy of Sciences of the United States*
641 *of America*, 87, 24, 9868–9872
- 642 Operto, G., Bulot, R., Anton, J.-L., and Coulon, O. (2006), Anatomically informed convolution kernels
643 for the projection of fMRI data on the cortical surface, in Proceedings 9th International Conference
644 on Medical Image Computing and Computer Assisted Intervention (Springer Verlag, Copenhagen,
645 Denmark), LNCS 4191, 300–307

- 646 Owens, J., Houston, M., Luebke, D., Green, S., Stone, J., and Phillips, J. (2008), Gpu computing,
647 *Proceedings of the IEEE*, 96, 5, 879–899
- 648 Pinel, P., Thirion, B., Mriaux, S., Jobert, A., Serres, J., Le Bihan, D., et al. (2007), Fast reproducible
649 identification and large-scale databasing of individual functional cognitive networks, *BMC Neurosci.*,
650 8, 1, 91
- 651 Risser, L., Vincent, T., Forbes, F., Idier, J., and Ciuciu, P. (2011), Min-max extrapolation scheme for fast
652 estimation of 3D Potts field partition functions. application to the joint detection-estimation of brain
653 activity in fMRI., *Journal of Signal Processing Systems*, 65, 3, 325–338
- 654 Thyreau, B., Thirion, B., Flandin, G., and Poline, J.-B. (2006), Anatomico-functional description of the
655 brain: a probabilistic approach, in Proceedings 31th Proceedings of the International Conference on
656 Acoustic, Speech and Signal Processing, volume V (Toulouse, France), volume V, 1109–1112
- 657 Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., et al. (2002),
658 Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of
659 the MNI MRI single-subject brain., *Neuroimage*, 15, 1, 273–89
- 660 Vincent, T., Ciuciu, P., and Thirion, B. (2008), Sensitivity analysis of parcellation in the joint detection-
661 estimation of brain activity in fMRI, in 5th International Symposium on Biomedical Imaging (Paris,
662 France), 568–571
- 663 Vincent, T., Risser, L., and Ciuciu, P. (2010), Spatially adaptive mixture modeling for analysis of within-
664 subject fMRI time series, *IEEE Transactions on Medical Imaging*, 29, 4, 1059–1074
- 665 Vincent, T., Warnking, J., Villien, M., Krainik, A., Ciuciu, P., and Forbes, F. (2013), Bayesian Joint
666 Detection-Estimation of cerebral vasoreactivity from ASL fMRI data, in 16th Proceedings Proc.
667 MICCAI, LNCS Springer Verlag, volume 2 (Nagoya, Japan), volume 2, 616–623
- 668 Yan, L., Zhuo, Y., Ye, Y., Xie, S. X., An, J., Aguirre, G. K., et al. (2009), Physiological origin of low-
669 frequency drift in blood oxygen level dependent (bold) functional magnetic resonance imaging (fMRI),
670 *Magnetic Resonance in Medicine*, 61, 4, 819–827