HAL
open science

Cadre bayésien markovien pour l'estimation de la HRF et la détection des activations en IRM fonctionnel

Alexandre Janon

To cite this version:

Alexandre Janon. Cadre bayésien markovien pour l'estimation de la HRF et la détection des activations en IRM fonctionnel. Traitement du signal et de l'image [eess.SP]. 2009. hal-00831300

HAL Id: hal-00831300

https://inria.hal.science/hal-00831300v1

Submitted on 6 Jun 2013

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.

Cadre bayésien markovien pour l'estimation de la HRF et la détection des activations en IRM fonctionnel.

Rapport du stage de Master recherche 2ème année Mathématiques appliquées effectué à l'INRIA Rhône-Alpes et au G.I.N. (Grenoble Institut des Neurosciences) sous la direction de Florence Forbes et Michel Dojat.

Alexandre Janon

Août 2009

Table des matières

1 Introduction **3**
1.1 Résumé 3
1.2 Présentation des équipes d'accueil 3
1.3 Présentation de l'imagerie par résonance magnétique 4
1.4 Problèmes posés par le traitement des données IRMf 6
1.4.1 Quantité de données à traiter 6
1.4.2 Bruit 6
1.4.3 Faiblesse de l'effet BOLD 7
1.4.4 Filtrage du signal 7

2 Analyse des signaux en IRMf **9**
2.1 Modélisation du signal (GLM) 9
2.2 Contrastes entre conditions 10
2.3 Approche standard par tests 10
2.4 Lissage spatial 11
2.5 Approche bayésienne 12
2.5.1 Champs de Markov 13
2.5.2 Inférence 14
2.6 Modélisation et estimation de la HRF 16

3 Première approche : modèle Gamma-Gamma-Gaussien et ICM **19**
3.1 Modélisation bayésienne 19
3.2 Estimation des paramètres 22
3.3 Inférence ICM 23
3.3.1 Algorithme d'inférence 23
3.3.2 Mise à jour de h 23
3.3.3 Mise à jour de a 25
3.3.4 Mise à jour de z 27
3.3.5 Initialisation 27
3.3.6 Algorithme final 29
3.4 Simulation de données 29
3.5 Evaluation de l'approche 30
3.5.1 Evaluation de l'estimation/détection 30
3.5.2 Evaluation de l'estimation de la HRF 36
3.5.3 Evaluation sur des données réelles 43

4 Seconde approche : modèle mélange de gaussiennes et EM variationnel **47**
4.1 Loi a posteriori des z 47
4.2 Loi a posteriori des a 50
4.3 Comparaison de conditions 51
4.3.1 Contraste 51
4.3.2 Divergence KL 51
4.4 Estimation de h 51
4.5 Estimation de β_1 53

5 Troisième approche : modèle auto-gaussien **53**

6 Conclusion et perspectives **54**
6.1 Conclusion 54
6.2 Extensions possibles 54
6.2.1 Meilleure prise en compte du bruit 54
6.2.2 Estimation de la HRF par zones 54

A Première implémentation **57**
A.1 Description 57
A.2 Format de fichier de design 58
A.2.1 Paramètres 58
A.2.2 Conditions 59
A.2.3 Exemples de fichier de design 59

B Deuxième implémentation **59**

1 Introduction

1.1 Résumé

Dans ce mémoire, nous présentons d'abord brièvement l'imagerie par résonance magnétique fonctionnelle, et nous dégageons les difficultés posées par le traitement statistique automatisé des images obtenues par cette technique. Puis nous présentons quelques unes des méthodes existantes, se basant sur un modèle fixe de la HRF, pour un tel traitement. La contribution de ce mémoire consiste en la proposition d'une méthode bayésienne pour estimer la HRF et prendre en compte la corrélation spatiale des activations *via* un modèle markovien. Après la description de cette méthode, nous l'évaluons sur des données simulées et réelles, et nous terminons en présentant des axes possibles d'extension.

1.2 Présentation des équipes d'accueil

Ce stage de recherche s'est déroulé au sein de deux équipes de recherche :

- l'équipe MISTIS, dépendant de l'INRIA Rhône-Alpes, dont l'objectif est de développer des méthodes statistiques adaptées à l'étude de phénomènes, de modèles et de données complexes, avec pour orientations applicatives privilégiées le traitement d'images et de données spatiales dans les domaines biomédicaux et industriels ; l'approche de l'équipe est basée sur l'introduction de la notion de structure dans les modèles et dans les données ; les thèmes de recherche sont les suivants : les modèles de mélange, les modèles markoviens, et les méthodes semi et non-paramétriques.
- l'équipe 5 (Neuro-imagerie Fonctionnelle et Métabolique) de l'Institut des Neurosciences de Grenoble, qui s'intéresse aux applications biomédicales *in vivo* de la résonance magnétique nucléaire (RMN). Les travaux, effectués tant sur l'homme que sur petit animal (rat, souris), visent au développement, à l'évaluation et à l'exploitation du potentiel en neurosciences cliniques, biologiques et cognitives de l'ensemble des méthodes de neuroimagerie par RMN. Les travaux ont été développés autour de trois thèmes de recherche : la micro-vascularisation cérébrale, le métabolisme cérébral et l'IRM fonctionnelle (IRMf) des activations cérébrales.

1.3 Présentation de l'imagerie par résonance magnétique

L'IRM (imagerie par résonance magnétique) est une technique non invasive permettant d'obtenir une image en trois dimensions d'une partie du corps. Elle est basée sur la mesure de la réponse de la zone étudiée à l'application d'un champ magnétique de forte intensité (actuellement entre 2 et 4 T), la distinction entre deux milieux différents étant fondée sur leurs réponses différentes.

Cette technique peut être utilisée pour produire une vue 3D du cerveau où sont distingués matière grise, matière blanche et liquide céphalo-rachidien, avec une résolution de l'ordre du millimètre ; on parle alors d'IRM anatomique cérébrale. Voir en Figure 1 pour un exemple d'image obtenue.

Une autre application de l'IRM, plus récente, est la mesure de l'activité du cerveau au cours du temps. Elle se base sur la différence de réponse magnétique entre une molécule d'hémoglobine oxygénée et une désoxygénée. Lorsqu'un neurone est activé, l'afflux sanguin autour de ce neurone augmente, ce qui se traduit par l'augmentation de la concentration en hémoglobine oxygénée et donc par une modification du signal IRM observé. Ce principe se nomme *effet BOLD* (pour blood-oxygen-level-dependent) et est à la base de l'IRM fonctionnelle (en abrégé IRMf). En IRMf, les acquisitions sont répétées au cours du temps ; la durée entre deux acquisitions étant le *temps de répétition* (TR), de l'ordre de deux à trois secondes ; voir Figure 2. Afin d'avoir un rapport signal-sur-bruit suffisant, une résolution spatiale de l'ordre de 3 millimètres est utilisée.

La conception d'une expérience d'IRMf consiste à définir différentes *conditions* qui sont répétées successivement. Ces conditions sont la réalisation de différentes tâches motrices ou cognitives (bouger les doigts de la main droite, comparaison d'objets...) ou encore une condition « nulle » (de contrôle). Par exemple, lors d'une expérience sur la vision des couleurs, on peut alterner la présentation d'une image en niveaux de gris (condition A), la présentation d'une image en couleurs (condition B) ou une condition C de repos (condition nulle). Chaque condition peut être présentée plusieurs fois au cours d'une session, et durant un temps plus ou moins long ; par exemple la condition A peut être présentée pendant 5 secondes, puis la B pendant 3 secondes, la C pendant 10 secondes, ensuite la B pendant 7 secondes, etc. La description des intervalles de temps durant lesquels chaque condition est active constitue le *design* de l'expérience.

L'analyse fonctionnelle des données issues de l'expérience consiste alors en la détermination des zones du cerveau activées lors de chacune des différentes conditions, et la comparaison des activations correspondant aux condi-

FIG. 1: Un exemple de coupe obtenue par IRM anatomique.

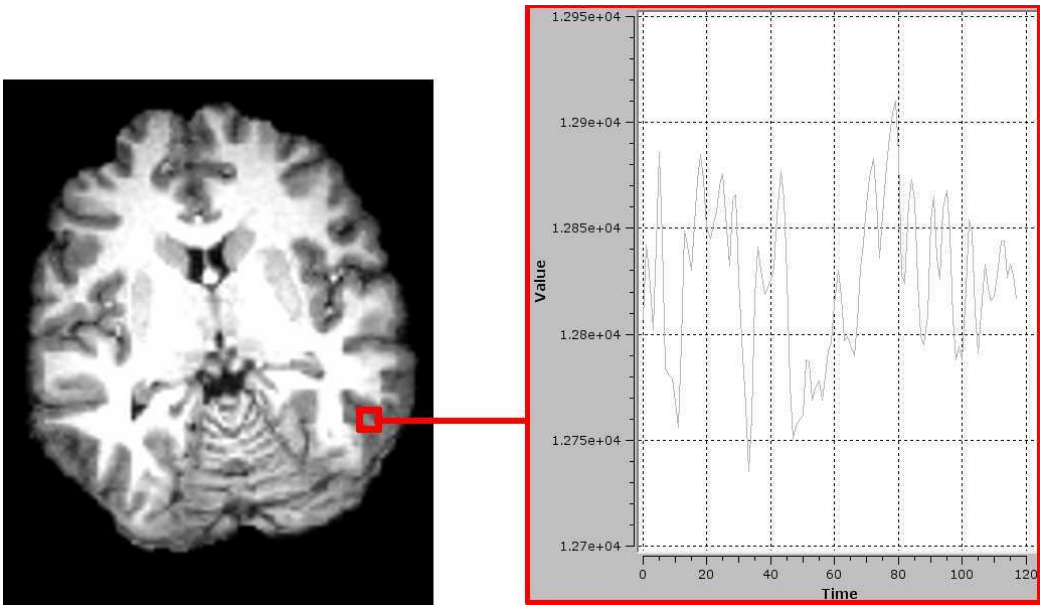


FIG. 2: Les données IRMf sont en quatre dimensions : la zone étudiée est découpée en petits cubes (d'environ 3 millimètres de côté) appelés *voxels* ; en chaque voxel on dispose d'un décours temporel (à droite) représentant l'évolution du signal BOLD mesuré au cours du temps.

tions entre elles (contraste) : par exemple, dans l'expérience sur la vision des couleurs, on peut vouloir déterminer les zones impliquées dans la vision en général (conditions A et B contre condition C) ou les zones actives dans la vision des couleurs (condition A contre B).

Ces expériences permettent d'améliorer la compréhension du fonctionnement du cerveau ; elles ont également des applications cliniques, telles que la délimitation des aires visuelles ou motrices pour préparer une intervention chirurgicale.

1.4 Problèmes posés par le traitement des données IRMf

1.4.1 Quantité de données à traiter

De part leur nature quadri-dimensionnelle, les données recueillies lors d'une expérience d'IRMf sont relativement volumineuses. A titre d'exemple réaliste, considérons une session de 360 secondes comportant une acquisition toutes les 3 secondes, et une zone d'acquisition étant un pavé de dimensions 20 cm, 20 cm et 10 cm, ainsi qu'une résolution spatiale de 3 millimètres ; si l'on suppose que les mesures sont stockées sous forme de *float* (2 octets), les données à traiter, pour un seul sujet, ont une taille de

$$\frac{360 \times 200 \times 200 \times 100}{3 \times 3 \times 3 \times 3} \times 2 \approx 34 \text{ Mo.}$$

Cette taille est à multiplier par le nombre de sujets, une étude cognitive étant rarement menée sur un seul sujet.

L'importance du volume des données impose l'utilisation de techniques spécifiques pour effectuer le traitement des données.

1.4.2 Bruit

Le signal recueilli en IRM est bruité de manière importante, les sources de bruit incluant :

- le bruit généré par la machine (bruit thermique, manque d'homogénéité de l'aimant produisant le champ magnétique...) ;
- le bruit physiologique, dû aux artefacts cardio-respiratoires ;
- le bruit « cognitif », dû aux sources de « distraction » du patient durant l'expérience (stimuli indésirables, endormissement du patient...) ;

A signaler également, même s'ils ne sont pas à classer dans les « bruits », les artefacts générés par les pré-traitements appliqués sur les données brutes, tels que : *slice correction* (correction du fait que les différentes coupes ne sont pas acquises au même moment) ou la correction des mouvements du patient.

1.4.3 Faiblesse de l'effet BOLD

L'afflux sanguin produit une variation du signal IRM mesuré mais celle-ci est relativement faible (de l'ordre de 1 à 2 %) par rapport au signal mesuré « au repos » (sans afflux sanguin). En d'autres termes la présence de molécules d'oxyhémoglobine dûe à une activation spécifique contribue peu au signal observé.

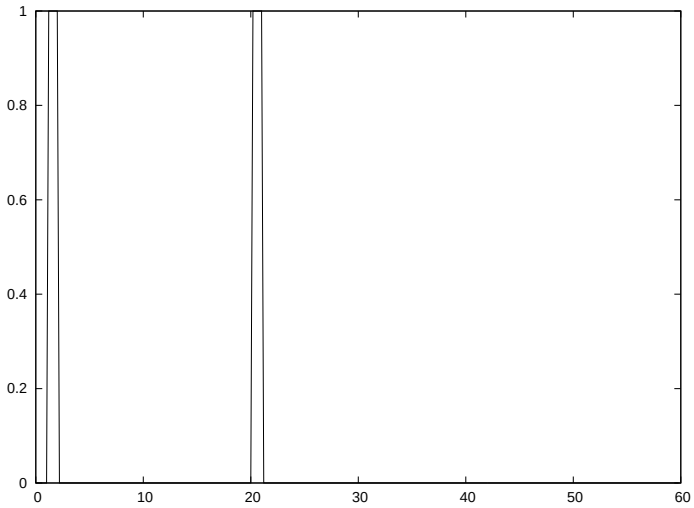
1.4.4 Filtrage du signal

Lors de l'activation d'un neurone, le débit sanguin ne passe pas brusquement du niveau zéro au niveau maximal, mais augmente progressivement ; un délai d'environ 5 à 6 secondes est nécessaire pour qu'il atteigne le niveau maximal. De même, lors d'une désactivation, le débit met un temps important à redescendre au niveau de repos.

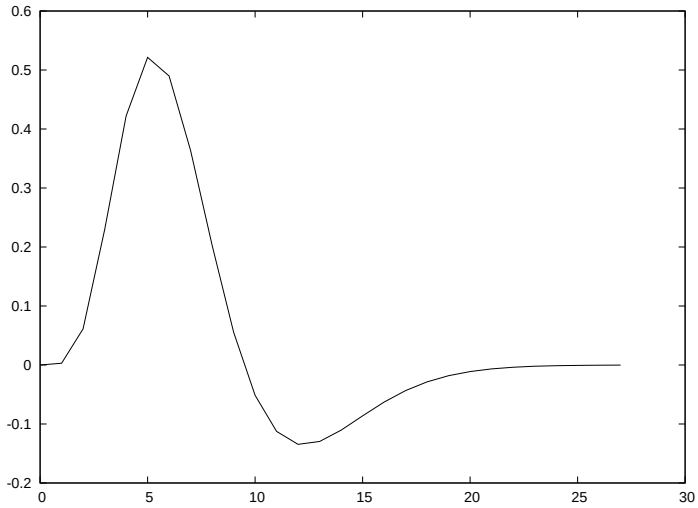
Dans un voxel activé par une condition donnée, le signal BOLD attendu est donc la convolution de la fonction indicatrice de la condition avec un filtre appelé *fonction de réponse hémodynamique* (HRF).

La fonction de réponse hémodynamique n'est pas connue ; de plus elle varie selon les individus, et suivant la zone du cerveau, au sein d'un même individu. Cependant elle reste toujours, pour un sujet adulte et en bonne santé, relativement proche de la HRF canonique présentée en Figure 3. Pour des sujets très jeunes ou pathologiques (par exemple les patients ayant subi un accident vasculaire cérébral, ou ceux souffrant d'épilepsie), la fonction de réponse hémodynamique peut être assez éloignée de cette HRF canonique.

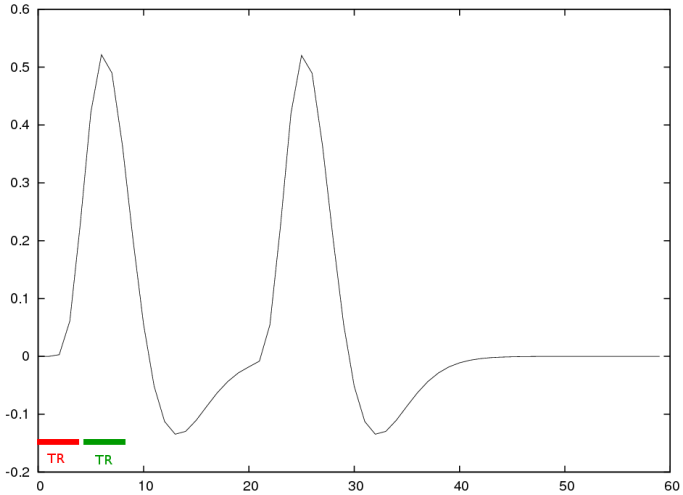
Pour plus de généralités sur les signaux IRMf, on pourra se reporter à l'ouvrage [13].



(a) Fonction indicatrice d'une condition expérimentale.



(b) Fonction de réponse hémodynamique.



(c) Signal BOLD attendu dans une zone activée par cette condition.

FIG. 3

2 Analyse des signaux en IRMf

2.1 Modélisation du signal (GLM)

Le signal IRM mesuré est modélisé par une collection de vecteurs $y_i = (y_{i1}, \ldots, y_{iT})$, $i = 1, \ldots, J$ (où J est le nombre de voxels composant l'image traitée et T la durée de l'expérience), vérifiant l'équation suivante, appelée *GLM* pour *General Linear Model* :

$$\forall i = 1, \ldots, J \;\; y_i = \sum_{m=1}^{M} a_{im}(x_m \star h) + n_i \mathbf{1} + \epsilon_i \tag{1}$$

où :

- l'indice m désigne une des conditions expérimentales, et M désigne le nombre de ces conditions ;
- le réel a_{im} est le *niveau de réponse* du voxel i à la condition m ;
- le vecteur x_m est le vecteur binaire indicateur des instants où la condition m est activée (*onsets*) : $x_{m,t} = 1$ si c'est le cas à l'instant t, et $x_{m,t} = 0$ sinon ;
- le vecteur h est un échantillonage de la fonction de réponse hémodynamique, de sorte que la convolution $x_m \star h$ contienne la « réponse attendue » illustrée en Figure 3 ; il est à noter que la période d'échantillonage de h peut être inférieure au temps de répétition TR, afin d'améliorer le calage des *onsets* qui ne sont pas forcément des multiples du TR ;
- le vecteur $\mathbf{1}$ est le vecteur de dimension T ne contenant que des 1, $n_i\mathbf{1}$ représente donc la « composante continue » du signal, qui n'est dûe à aucune des conditions prévues dans l'expérience ; certains modèles incluent de la même manière des composantes « basses fréquences » (*drifts*) dûes aux mouvements lents du patient, aux artefacts cardiovasculaires, à la respiration, ainsi qu'aux dérives de l'appareil d'acquisition (cf. [14], chapitre 2, §4.3).
- le vecteur ϵ_i est le vecteur des résidus ; le modèle le plus simple (bruit blanc gaussien) suppose $(\epsilon_i)_{i=1,\ldots,J}$ indépendants, suivant une loi normale $\mathcal{N}(0, \sigma^2 I_T)$ (où I_T désigne la matrice identité $T \times T$) ; cette hypothèse, peu réaliste, est souvent remplacée par celle d'une collection $(\epsilon_i)_{i=1,\ldots,T}$ de séries chronologiques indépendantes suivant chacune un modèle AR(1) gaussien (cf [26]).

L'équation 1 peut s'interpréter comme une régression linéaire multiple :

$$y_i = X\widetilde{a}_i + \epsilon_i$$

où y_i est la variable expliquée et la matrice X, supposée de plein rang $(M+1)$, la matrice des variables explicatives :

$$X = \begin{pmatrix} (x_1 \star h) & | & (x_2 \star h) & | & \dots & | & \mathbf{1} \end{pmatrix}$$

et $\widetilde{a}_i$ est le vecteur contenant les paramètres

$$\widetilde{a}_i = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{im} \\ n_i \end{pmatrix}$$

Lorsque X est connue, ces paramètres peuvent être estimés par la méthode des moindres carrés :

$$\widehat{\widetilde{a}_i} = (X^T X)^{-1} X^T y_i$$

$$\widehat{\sigma^2} = \frac{\sum_i \left|\left| y_i - X\widehat{\widetilde{a}_i} \right|\right|^2}{T - (M+1) - 1}$$

2.2 Contrastes entre conditions

La détermination des zones activées par une condition expérimentale en comparaison d'une autre se fait en calculant pour chaque voxel l'*effet*

$$\widehat{\gamma_i} = c^T \widehat{a_i}$$

où $a_i = (a_{i1}, \dots, a_{im})^T$ et c est le vecteur des contrastes, qui indique les conditions expérimentales que l'on veut comparer. Par exemple si l'on souhaite comparer les zones activées par la condition 1 par rapport à la condition 3, on prendra $c = (1, 0, -1)$; pour comparer la condition 1 avec la moyenne des conditions 2 et 3, on choisira $c = (-1, 0.5, 0.5)$.

2.3 Approche standard par tests

Une fois l'effet $\widehat{\gamma_i}$ calculé (phase d'*estimation*), il reste à déterminer si celui-ci est statistiquement significatif ou non (phase de *détection*). Pour cela on utilise un test statistique, visant à comparer l'hypothèse nulle « le voxel n'est pas activé » contre l'hypothèse « le voxel est activé ».

Un test fréquemment utilisé (test de Student), consiste à calculer une t-statistique :

$$t_j = \frac{\widehat{\gamma_i}}{\widehat{\sigma}\sqrt{c^T(X^TX)^{-1}c}}$$

qui, sous l'hypothèse nulle, suit une loi de Student à $M+1$ degrés de libertés (cf. [6], chapitre 7).

Les statistiques t_j étant calculées pour chaque voxel, on obtient une « SPM » (*statistical parameter map*) qui donne le niveau de significativité du contraste mesuré dans chaque voxel. Il faut alors choisir une valeur seuil en dessous de laquelle un voxel est considéré comme non significatif.

Le choix de cette valeur seuil est rendu difficile par le fait que des tests multiples sont effectués. En effet supposons que nous utilisions sur chaque voxel un test possédant un risque de première espèce (faux-positif) α ; si nous effectuons ce test n fois le nombre moyen de faux-positifs détectés sera de $n\alpha$. Par exemple supposons $\alpha = 0.05$ et $n = 10^4$ (nombre réaliste de voxels dans une image), il y aura en moyenne 50 faux-positifs détectés. Pour résoudre ce problème, une correction possible est la correction de Bonferroni, qui nécessite d'utiliser un test avec un risque de première espèce α/n pour obtenir un niveau de risque « global » majoré par α, cette borne pouvant être atteinte dans le cas où les tests sont indépendants. Cette correction est trop « conservative » dans le cas qui nous intéresse car les tests sont loin d'être indépendants : les activations ont lieu dans des zones contigües de plusieurs voxels, et le résultat d'un test sur un voxel est très corrélé avec les résultats des tests sur les voxels voisins. Pour tenir compte de cette dépendance, on doit abandonner les tests voxel par voxel et considérer la SPM dans son ensemble, en la modélisant comme un champ gaussien (*gaussian random field*, GRF) ; cf. [6], chapitre 14.

Pour plus de généralités sur l'approche GLM, on pourra se reporter à l'ouvrage [13].

2.4 Lissage spatial

Préalablement à l'estimation des paramètres dans l'équation 1, on applique généralement un lissage de l'image, afin d'augmenter le rapport signal-sur-bruit (SNR), au prix d'une détérioration de la résolution spatiale.

Une méthode de lissage classique est la convolution avec une gaussienne ; cependant ils existe d'autres lissages dits adaptatifs, qui tiennent mieux compte des contours de l'image, tels que l'algorithme PS (propagation-séparation), cf. [20] et [19].

2.5 Approche bayésienne

Par rapport à l'approche classique évoquée ci-dessus, une approche bayésienne présente les avantages suivants :

- une gestion « élégante » des incertitudes : en particulier une analyse bayésienne donne une distribution *a posteriori* qui possède une interprétation intuitive (à comparer avec l'interprétation plus délicate des niveaux de risques dans les tests et des intervalles de confiance en statistique classique) ;
- un cadre théorique bien adapté pour intégrer des connaissances *a priori* (par exemple anato-fonctionnelles) concernant le signal à traiter, améliorant la précision des résultats fournis ;
- des méthodes de choix de modèles (Bayes factor).

Pour plus d'informations sur l'apport des méthodes bayésiennes en IRMf, on pourra consulter [24].

Dans une approche bayésienne, on considère les paramètres $\Theta = (\theta_i)_i$ comme des variables aléatoires, dont on spécifie la loi $p(\Theta)$, qui modélise la connaissance sur les paramètres que l'on a avant d'observer les données, d'où son nom de loi a priori. On spécifie également une *vraisemblance* $p((y_k)_k|\Theta)$ qui modélise le comportement des données une fois les paramètres connus.

L'*inférence* bayésienne consiste à calculer (ou, tout au moins, connaître certaines caractéristiques, comme le(s) mode(s), la moyenne, la variance, ...) la distribution a posteriori $p(\Theta|(y_k)_k)$, qui décrit la connaissance que l'on a sur les paramètres une fois que l'on a observé les données. Cette distribution est donnée par la règle de Bayes :

$$p(\Theta|(y_k)_k) = \frac{p((y_k)_k|\Theta)p(\Theta)}{p((y_k)_k)} = \frac{p((y_k)_k|\Theta)p(\Theta)}{\int p((y_k)_k|(\theta_i)_i)d((\theta_i)_i)} \tag{2}$$

Un modèle purement bayésien considère *tous* les paramètres comme des variables aléatoires et leur donne une loi *a priori*, mais un modèle peut tout à fait continuer à traiter certains paramètres comme de « vrais » paramètres, sans leur donner de lois *a priori*. Ces derniers paramètres sont alors estimés de manière « classique », par exemple par maximum de vraisemblance.

Un inconvénient des méthodes bayésiennes est que l'intégrale figurant dans (2) est souvent incalculable analytiquement, et un calcul numérique approché est souvent infaisable, compte tenu de la dimension de cette intégrale (qui est égale aux nombre de paramètres). La loi *a posteriori* est dans ce cas connue à une constante multiplicative près, et il faut alors avoir recours à des méthodes particulières, dont certaines seront décrites en section 2.5.2.

2.5.1 Champs de Markov

Le signal mesuré en IRMf est très corrélé spatialement, puisque les zones activées s'étendent sur plusieurs voxels. Tenir compte de cette dépendance spatiale permet d'améliorer considérablement la détection et l'estimation des activations. Dans cette partie, nous décrivons une manière de tenir compte de la dépendance spatiale entre les voxels.

Champs de Markov et lois de Gibbs. Soit V un ensemble (fini) de points appelés *sites*. On se donne un graphe G non orienté, sans boucle, dont les sommets sont les éléments de V. Pour un site $v \in V$, on note $\mathcal{N}(v)$ l'ensemble des voisins de v, c'est à dire l'ensemble des sites reliés à v par une arête de G. Comme G n'a pas de boucle, un sommet ne peut être son propre voisin.

Une collection $(Z_v)_{v \in V}$ de variables aléatoires est un *champ de Markov* sur G si la loi conditionnelle de Z_v sachant $Z_{v'}$ pour $v' \neq v$ est égale à la loi conditionnelle de Z_v sachant $Z_{v'}$ pour $v' \in \mathcal{N}(v)$. Le graphe G donne donc l'information sur les interactions possibles entre les variables (Z_v).

Une *loi de Gibbs* sur G est une loi dont la densité se factorise suivant les cliques de G, i.e. une loi dont la densité p est de la forme :

$$p(z_1, \ldots, z_{|V|}) \propto \prod_{C \in \mathcal{C}(G)} V_C(z_v; v \in C) \tag{3}$$

où on a noté $\mathcal{C}(G)$ l'ensemble des cliques de G, une *clique* étant un ensemble maximal de sites deux à deux voisins. Les fonctions V_C sont appelées *potentiels de Gibbs*.

Il est facile de vérifier que si $(Z_v)_v$ suit une loi de Gibbs sur G, alors c'est un champ de Markov sur G. La réciproque est également vraie, si la densité de la loi de $(Z_v)_v$ est strictement positive (théorème de Hammersley-Clifford, cf. [3], [1]).

Modèle de Potts Un exemple important de loi de Gibbs est donné par le *modèle de Potts* dans lequel toutes les cliques sont d'ordre 2 et où chaque variable peut prendre un nombre fini de valeurs appelées *classes* et où pour chaque clique $C = \{v, v'\}$, supposée d'ordre au plus égal à 2, le potentiel de Gibbs est donné par

$$V_C(z_v, z_{v'}) = \exp(-2\beta\delta(z_v, z_{v'})) \tag{4}$$

soit encore

$$p((z_v)_v) \propto \exp\left(-\beta \sum_{v} \sum_{v' \in \mathcal{N}(v)} \delta(z_v, z_{v'})\right) \tag{5}$$

où δ désigne le complémentaire du symbole de Kronecker : $\delta(x, y) = 0$ si $x = y$ et $\delta(x, y) = 1$ sinon, et $\beta > 0$ est un coefficient rendant compte de la force des interactions entre les différents sites. Dans ce modèle, deux variables voisines ont une probabilité d'être égales d'autant plus importante que β est grand.

Modèle auto-gaussien Un autre exemple de loi de Gibbs, toujours avec des cliques d'ordre 2, mais avec cette fois-ci des variables continues, est le modèle auto-gaussien, où la loi conjointe des $(Z_v)_v$ est :

$$p((z_v)_v) \propto \exp\left(-\beta \sum_{v} \sum_{v' \in \mathcal{N}(v)} (z_v - z_{v'})^2\right) \tag{6}$$

Le coefficient $\beta > 0$ est toujours un coefficient traduisant la force de l'interaction entre deux variables « voisines ».

Modélisation de l'interaction spatiale entre les voxels. Lors du traitement bayésien d'une image IRMf, un modèle de Potts peut être utilisé comme *a priori* régularisant de la manière suivante : les sommets du graphe de dépendance sont les voxels eux-mêmes, les classes sont {activé, non activé} et les voisins d'un voxel sont ses 6 voisins au sens physique (en haut, en bas, à gauche, à droite, en avant, en arrière). L'équation (5) rend alors bien compte du fait que si un voxel est activé (resp. non activé), alors ses voisins ont une probabilité importante d'être activés (resp. non activés). Ce modèle permet donc d'effectuer simultanément la classification des voxels et la régularisation spatiale. Nous utilisons ce modèle dans nos deux premières approches.

Notre troisième approche ne classifie pas les voxels, mais impose une régularisation spatiale directement au travers du *prior* auto-gaussien utilisé sur les niveaux de réponse a_{im}, avec le même graphe de dépendence que ci-dessus.

2.5.2 Inférence

Dans cette partie nous présentons quelques méthodes d'inférence utilisées pour obtenir des informations sur une loi *a posteriori* $\Theta|y$ (Θ est l'ensemble des « paramètres »et y sont les données observées), dont la densité est connue seulement à une constante multiplicative près, ainsi que décrit dans la section 2.5.

Monte-Carlo par chaînes de Markov (MCMC) Les approches de type Monte-Carlo consistent en la simulation d'un échantillon de la loi *a posteriori*, puis du calcul des caractéristiques de cette loi (moyenne, variance, histogramme...) à partir de l'échantillon simulé.

L'algorithme utilisé pour la simulation de l'échantillon est l'algorithme de Metropolis-Hastings (cf. [7]), dont l'échantillonnage de Gibbs est un cas particulier. Cet algorithme construit, à partir de la donnée d'une loi dont la densité est connue à une constante près, une chaîne de Markov dont la distribution stationnaire est la loi à simuler.

Les méthodes de Monte-Carlo donnent de bonnes approximations, sont bien justifiées théoriquement et donnent accès à toutes les caractéristiques des lois mais demandent des temps de calcul généralement assez longs.

Iterated Conditional Modes (ICM) L'algorithme ICM (cf. [2]) peut être utilisé pour déterminer le mode d'une loi jointe $p(x_1, x_2, \ldots, x_n)$ qui n'est pas maximisable directement, mais dont les lois conditionnelles $p(x_i|x_j, i \neq j)$ le sont.

Le principe de l'algorithme est le suivant :

Initialiser $x_1, \ldots, x_n$;
Tant que non convergé faire :
Pour i entre 1 et n faire :
Mettre à jour x_i selon $x_i \leftarrow \arg\max_{\widetilde{x}_i} p(\widetilde{x}_i \| x_j, j \neq i)$;

L'algorithme ICM est plus rapide que la méthode MCMC mais il ne donne pas accès à toute l'information sur la loi *a posteriori* ; de plus il est seulement convergent vers un maximum local de la vraisemblance (la vraisemblance ne faisant qu'augmenter à chaque étape), et est donc assez sensible à l'initialisation de l'algorithme.

Notre première approche utilise cette méthode d'inférence.

Approches variationnelles, variational EM Dans une approche variationnelle, la loi a posteriori $p(\Theta|y)$ est approximée par une autre loi $q(\Theta)$; plus précisément nous imposons la « forme » de la loi q, cette loi dépendant de paramètres qui sont déterminés en cherchant à minimiser la divergence de Kullback-Leibler

$$KL = \mathbf{E}_q \left[\ln \frac{q(\Theta)}{p(\Theta|y)} \right],$$

qui est une manière de quantifier la « distance » entre les deux lois q et $p(.|y)$.

L'approche EM variationnelle (cf. [11] ou [22]), basée sur une version fonctionnelle de l'algorithme EM, applique ce principe en remplaçant la loi a

posteriori du modèle de Markov caché par une loi factorisée sur les sommets, permettant d'obtenir à la fois une approximation de la loi des paramètres *a posteriori*, et d'estimer les paramètres non bayésiens par maximum de vraisemblance.

Les méthodes EM variationnelles sont déjà utilisées dans certaines approches de traitement des données issues d'IRM fonctionnelles (cf. [18]).

Nous utiliserons une méthode variationnelle dans notre seconde approche.

2.6 Modélisation et estimation de la HRF

Dans l'approche classique présentée plus haut, la fonction HRF est posée comme un *a priori* fixe, la détection et l'estimation des activations étant menées comme si tous les sujets avaient la même HRF dans toutes les parties du cerveau.

Il existe plusieurs formules analytiques modélisant une HRF ayant un sens physiologique. Nous donnerons comme exemple la « différence de deux gamma » :

$$h(t) = \left(\frac{t}{d_1}\right)^{a_1} \exp\left(-\frac{t-d_1}{b_1}\right) - c\left(\frac{t}{d_2}\right)^{a_2} \exp\left(-\frac{t-d_2}{b_2}\right)$$

avec t le temps exprimé en secondes. Les valeurs canoniques données aux paramètres sont les suivantes (cf. [20]) $a_1 = 6$, $a_2 = 12$, $b_1 = b_2 = 0.9$, $d_i = a_i b_i$ pour $i = 1, 2$ et $c = 0.35$. Voir en Figure 4 pour une représentation de cette fonction.

On peut également vouloir estimer la HRF du sujet à partir des mesures effectuées et l'utiliser pour la détection/estimation des activations. Pour un sujet sain, cette estimation supplémentaire permet d'espérer « raffiner » la détection des activations ; pour un sujet très jeune ou atteint d'une maladie, cette estimation est nécessaire si l'on veut traiter l'image, compte tenu de la forme très différente de la HRF chez ces sujets.

Des approches paramétriques d'estimation de la HRF ont été proposées, avec estimation des paramètres par maximum de vraisemblance (ou maximum *a posteriori*) (cf. [25]) ; il existe également des approches semi-paramétriques, où la fonction HRF est décrite comme une combinaison linéaire (avec d'éventuelles contraintes sur les coefficients pour que la HRF estimée fasse sens physiologiquement) de fonctions de base ; enfin on peut également faire une estimation non-paramétrique. Dans ce cas, pour améliorer l'estimation de la HRF h on peut intégrer deux connaissances *a priori* (cf. [15]) :

- la fonction h vérifie $h(0) = 0$ et que $h(D) = 0$ pour D de l'ordre de 25 à 30 secondes [1] ;

[1] même si dans certains cas pathologiques le retour à zéro peut être plus long, cf. [4]

– la fonction h est à variations lisses, c'est-à-dire que sa dérivée seconde a une norme L^2 « petite ».

A noter également que lors de l'estimation de la HRF un problème d'identifiabilité se pose : en effet, dans le modèle linéaire (1), si on multiplie la fonction h par un facteur constant $k \neq 0$ et que l'on divise les niveaux de réponse a par le même facteur k, on obtient deux valeurs des variables à estimer qui maximisent la vraisemblance du modèle (ou la densité *a posteriori*). On peut résoudre ce problème en normalisant h, soit en imposant $||h||^2 = 1$, ce qui devrait rendre le problème bien posé (cf. [15]), au signe de h près ; ou encore en fixant le maximum de $|h|$ à 1.

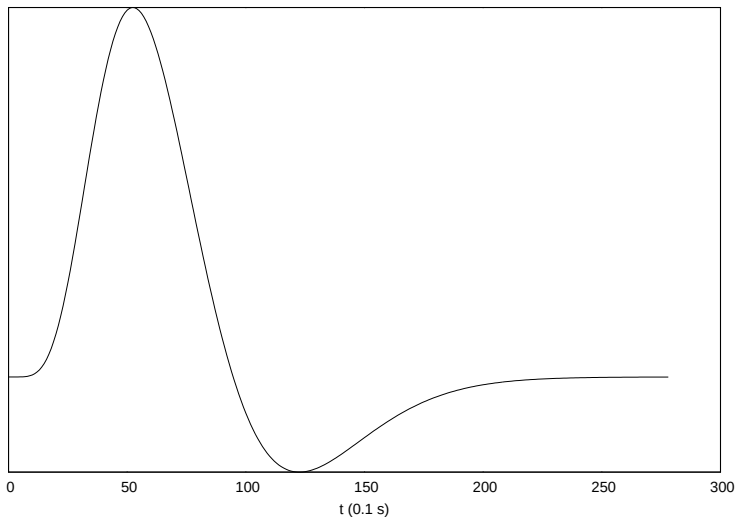


FIG. 4: Profil de la fonction de réponse hémodynamique canonique.

3 Première approche : modèle Gamma-Gamma-Gaussien et ICM

Nous avons choisi de partir du modèle présenté dans [15]. Nous nous proposons de modifier ce modèle en enlevant les hyperparamètres bayésiens, et d'utiliser à la place un champ de Markov afin de modéliser la dépendance spatiale entre les voxels. Les auteurs du modèle original utilisent une méthode de type MCMC pour effectuer l'inférence, ce qui donne des temps de calcul assez longs. Nous utilisons un algorithme de type ICM afin de disposer d'une méthode de traitement plus rapide.

Cette approche a été étudiée indépendamment de l'article [23], cette dernière approche utilisant plus de priors et sa méthode d'inférence étant encore un MCMC.

3.1 Modélisation bayésienne

Nous reprenons les notations de l'équation (1) :

$$y_i = \sum_{m=1}^{M} a_{im}(X_m h) + n_i \mathbf{1} + \epsilon_i,\ i = 1, \ldots, J \tag{7}$$

où X_m désigne l'application linéaire $h \mapsto x_m \star h$.

Pour comprendre la forme de cette matrice X_m, voici l'exemple (peu réaliste) d'un expérience de 6 secondes, avec un TR de 2 secondes ; supposons que le temps de retour à zéro de h soit de 3 secondes, et que nous échantillonions la fonction de réponse hémodynamique toutes les 0.5 secondes ; le vecteur h est alors de taille 6. Considérons une condition expérimentale

apparaîssant pour $t = 0$ s et $t = 4$ s. La matrice X_m est alors égale à :

$$X_m = \left(\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)$$

La matrice X_m est donc formée d'une succession blocs « identité » quand la condition est active ou « nul » sinon.

Nous faisons l'hypothèse d'un bruit blanc gaussien : les $(\epsilon_i)_i$ sont indépendants et suivent une loi normale centrée, de matrice de variance/covariance $\sigma_e^2 I_T$, de sorte que $y_i | a_i, h$ suit une loi normale, de moyenne

$$\sum_{m=1}^{M} a_{im}(X_m h) + n_i \mathbf{1}$$

et de variance $\sigma_e^2 I_T$.

Nous supposons donc connues les matrices X_m (rentrées par l'utilisateur) et les observations y_i. Sont à estimer h, a_{im}, n_i et σ_e^2.

On introduit une « étiquette » z_{im} définie comme suit :

$$z_{im} = \begin{cases} +1 & \text{si le voxel } i \text{ est activé dans la condition } m \\ -1 & \text{si le voxel } i \text{ est désactivé dans la condition } m \\ 0 & \text{si le voxel } i \text{ est neutre dans la condition } m \end{cases}$$

Ces étiquettes ne sont pas directement observées (ce sont donc des données manquantes), elles sont introduites artificiellement, afin de modéliser le niveau de réponse a_{im} comme dépendant de z_{im}.

Pour chaque m, les variables $(a_{im})_i$ sont supposées former une famille de variables indépendantes conditionnellement aux $(z_{im})_i$ et pour chaque i, la famille $(a_{im})_m$ est une famille de v.a. indépendantes, de même que $(z_{im})_m$ (on suppose donc qu'il n'y a pas d'influence entre deux conditions expérimentales différentes).

La vraisemblance complète du modèle à estimer s'écrit

$$p(y, a, h, z) = p(y|a, h)p(a|z)p(h)p(z)$$

soit, compte tenu des hypothèses d'indépendance formulées plus haut :

$$p(y, a, h, z) = \prod_i \left(p(y_i|a_i, h) \prod_m p(a_{im}|z_{im}) \right) \times p(h) \times \prod_m p((z_{\cdot m}))$$

Décrivons maintenant chacun des termes de cette factorisation. Dans le premier modèle que nous proposons (modèle « Gamma-Gamma-Gaussien »), la variable a_{im} sachant z_{im} est supposée suivre une des lois :

$$a_{im}|z_{im} \sim \begin{cases} \text{Gamma}(\alpha_{1m}, \beta_{1m}) & \text{si } z_{im} = 1 \\ \text{OppGamma}(\alpha_{-1m}, \beta_{-1m}) & \text{si } z_{im} = -1 \\ \text{Normale}(0, \sigma_{0m}^2) & \text{si } z_{im} = 0 \end{cases}$$

où $\text{Gamma}(\alpha, \beta)$ est une loi de densité

$$g_{\alpha,\beta}(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \mathbb{1}_{\mathbb{R}_+}$$

et où une variable X est dite suivre une loi $\text{OppGamma}(\alpha, \beta)$ si $-X$ suit une loi $\text{Gamma}(\alpha, \beta)$.

La loi *a priori* sur les $(z_{im})_i$ est celle d'un modèle de Potts (cf 2.5.1) sur l'ensemble des voxels :

$$p((z_{im})_i) \propto \exp\left(-\beta_1 \sum_{i=1}^{J} \sum_{j \in \mathcal{N}(i)} \delta(z_{im} - z_{jm}) \right)$$

où $\beta_1 > 0$ est un coefficient à définir par l'utilisateur ; la famille $(z_{im})_m$ est supposée indépendante quelque soit i. Nous avons donc M modèles de Potts (un par condition expérimentale) que nous supposons indépendants.

L'estimation de la fonction de réponse hémodynamique h est non paramétrique ; cependant la loi *a priori* sur h tient compte du fait que cette fonction est à variations lisses, i.e. sa dérivée seconde n'est pas trop « grande » en valeur absolue, on choisit ainsi (comme dans [15]) :

$$p(h) \propto \exp\left(-\beta_2 h^T S h\right)$$

avec $S = D_2^T D_2$ et D_2 la matrice de l'application qui à un vecteur associe ses différences finies secondes :

$$D_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

de sorte que la loi *a priori* sur h assure la contrainte de variations lisses en rendant « improbable » les h ayant une norme de la dérivée seconde importante ; β_2 est un second coefficient fixé par l'utilisateur, qui détermine l'importance donnée à cette contrainte.

Les contraintes $h(0) = h(D) = 0$ et de normalisation ($||h||_{\ell^2}^2 = 1$) seront assurées de manière « forte » lors de l'estimation de h (cf. ci-après).

3.2 Estimation des paramètres

Le modèle décrit précédemment comprend plusieurs paramètres, estimés de manière « classique » (sans loi *a priori*) :

- n_i est estimé de manière à ce que les résidus ϵ_i aient une moyenne nulle ; soit :
$$\widehat{n_i} = \frac{1}{T} \sum_{t=1}^{T} \left(y_{it} - \sum_{m=1}^{M} a_{im} (X_m h)_t \right)$$
- σ_e^2 est estimée par maximum de vraisemblance :
$$\widehat{\sigma_e^2} = \frac{1}{T \times J} \sum_{j=1}^{J} \sum_{t=1}^{T} (y_{it} - \widehat{n_i})^2$$
(nous n'appliquons pas de correction du biais car celle-ci est négligeable compte tenu des ordres de grandeurs de T et J).
- σ_{0m}^2 également :
$$\widehat{\sigma_{0m}^2} = \frac{1}{N_0} \sum_{i | z_{im} = 0} a_{im}^2$$
où $N_0 = \text{Card } \{i | z_{im} = 0\}$.
- Les paramètres $\alpha_{\pm 1m}$ et $\beta_{\pm 1m}$ sont estimés en utilisant les estimateurs de Thom pour une loi Gamma (cf. [10]), qui s'expriment ainsi, pour un

échantillon de moyenne arithmétique m_a et de moyenne géométrique m_g :

$$\begin{cases} \alpha' &= \dfrac{1}{4R}\left(1+\sqrt{1+\dfrac{4}{3}R}\right) \\ \widehat{\alpha} &= \alpha' + \kappa \\ \widehat{\beta} &= \dfrac{\widehat{\alpha}}{\mu} \end{cases}$$

avec

$$R = \ln(m_a) - \ln(m_g)$$

et

$$\kappa = \begin{cases} 0 & \text{si } \alpha' < 0.9 \\ 0.0092 + \dfrac{\alpha'-1}{24-96\alpha'} & \text{sinon.} \end{cases}$$

3.3 Inférence ICM

3.3.1 Algorithme d'inférence

Nous utilisons des itérations ICM, la mise à jour d'un paramètre $\theta \in \{a, h, z\}$ se faisant en maximisant (le logarithme de) $p(\theta|y, \text{autres paramètres})$ et en itérant sur le paramètre θ à mettre à jour.

3.3.2 Mise à jour de h

Le vecteur h est choisi de façon à maximiser $\ln(p(h|y,a))$; or :

$$\begin{aligned} \ln p(h|y,a) &= \text{Cste} + \frac{-1}{2\sigma_e^2}\left(\sum_i \left|\left| y_i - \left(\sum_m a_{im}X_m\right)h - n_i\mathbf{1}\right|\right|^2\right) - \beta_2 h^T S h \\ &= \text{Cste} + \frac{-1}{2\sigma_e^2}\left(\sum_i ||c_i - M_i h||^2\right) - \beta_2 h^T S h \end{aligned}$$

avec

$$\begin{cases} M_i &= \sum_m a_{im}X_m \\ c_i &= y_i - n_i\mathbf{1} = \epsilon_i + M_i h \end{cases}$$

On a :

$$\sum_i ||c_i - M_i h||^2 = \text{Cste} + h^T\left(\sum_i M_i^T M_i\right)h - 2h^T\sum_i M_i^T c_i$$

D'où

$$\begin{aligned}
\ln p(h|y,a) &= \text{Cste} + \frac{-1}{2\sigma_e^2} h^T \left(\sum_i M_i^T M_i \right) h + \frac{1}{\sigma_e^2} h^T \sum_i M_i^T c_i - \beta_2 h^T S h \\
&= \text{Cste} + h^T \left[\frac{-1}{2\sigma_e^2} \sum_i M_i^T M_i - \beta_2 S \right] h + h^T \left[\frac{1}{\sigma_e^2} \sum_i M_i^T c_i \right] \\
&= \text{Cste} + h^T Q h + h^T v
\end{aligned}$$

en posant :

$$\begin{cases} Q &= \frac{-1}{2\sigma_e^2} \sum_i M_i^T M_i - \beta_2 S, \text{ matrice symétrique définie négative} \\ v &= \frac{1}{\sigma_e^2} \sum_i M_i^T c_i \end{cases}$$

Nous voulons donc maximiser $h^T Q h + h^T v$ par rapport à h, sous les contraintes $h_1 = e_1^T h = 0$ (le symbole ^{T} désignant la transposée) et $h_D = e_D^T h = 0$. Le théorème des multiplicateurs de Lagrange donne une condition nécessaire, à savoir l'existence de réels μ et ν tels que

$$2Qh + v + \mu e_1 + \nu e_D = 0 \tag{8}$$

En faisant le produit scalaire de l'équation (8) avec le vecteur e_1 on trouve :

$$2e_1^T Q h + e_1^T v + \mu = 0 \tag{9}$$

soit

$$\mu = -(2e_1^T Q h + e_1^T v) \tag{10}$$

De même, en faisant le produit scalaire de (8) avec e_D on trouve :

$$\nu = -(2e_D^T Q h + e_D^T v) \tag{11}$$

Les équations (linéaires) nous donnant la mise à jour de h sont donc :

$$\begin{cases} h_1 & = 0 \\ h_D & = 0 \\ 2Qh + v - (2e_1^T Q h + e_1^T v) e_1 - (2e_D^T Q h + e_D^T v) e_D & = 0 \end{cases} \tag{12}$$

Une fois h calculé en résolvant ce système linéaire, il est lissé à l'aide d'un filtre gaussien. Plus précisément h est remplacé par Gh, où G est la matrice définie par :

$$G_{i,j} = \exp\left(\frac{(i-j)^2}{BW} \right)$$

où $BW > 0$ est un paramètre qui contrôle la « largeur » de la gaussienne utilisée pour filtrer ; plus BW est grand, plus le lissage est important.

Enfin h est normalisé, c'est à dire qu'il est remplacé [2] par

$$\frac{1}{||h||_{\ell^2}} h.$$

Nous avons constaté qu'en estimant h de cette manière, nous obtenons quelquefois l'opposé de la HRF réelle ; c'est un effet dû au problème d'identifiabilité que nous avons mentionné en 2.6. Pour régler ce problème, nous calculons la valeur moyenne de h ; si celle-ci est négative c'est que nous avons trouvé l'opposé de la HRF réelle et h est alors remplacé par $-h$.

3.3.3 Mise à jour de a

Pour maximiser $p(a|z,h,y)$, il suffit, par indépendance des $(a_{im})_m$, de maximiser $p(a_i|z,h,y)$ pour chaque i fixé.

Or

$$\begin{aligned}
\ln(p(a_i|z,h,y) &= \text{Cste} + \ln(p(a_i|z_i)) + \ln(p(y_i|a_i,h)) \\
&= \text{Cste} + \frac{-1}{2\sigma_e^2} \left|\left| c_i - \sum_{m=1}^{M} a_{im} X_m h \right|\right|^2 \\
&+ \sum_{m/z_{im}=1} (\alpha_{1m} - 1) \ln a_{im} - \beta_{1m} a_{im} \\
&+ \sum_{m/z_{im}=-1} (\alpha_{-1m} - 1) \ln(-a_{im}) + \beta_{-1m} a_{im} \\
&+ \sum_{m/z_{im}=0} \frac{-1}{2\sigma_{0m}^2} a_{im}^2
\end{aligned}$$

avec

$$c_i = y_i - n_i \mathbf{1}$$

Avant de dériver $\ln(p(a_i|z_i,h,y_i))$ par rapport à a_{im} effectuons un calcul pré-

[2] La contrainte $h^T h = 1$ peut également s'intégrer dans les contraintes du problème d'optimisation, mais cela conduit à un système non linéaire.

liminaire :

$$\begin{aligned}
\left|\left|c_i - \sum_m a_{im} X_m h\right|\right|^2 &= ||c_i||^2 - 2\sum_m a_{im} < X_m h, c_i > + \left|\left|\sum_m a_{im} X_m h\right|\right|^2 \\
&= \text{Cste} - 2\sum_m a_{im} < X_m h, c_i > + \sum_m a_{im}^2 \, ||X_m h||^2 \\
&\quad + \sum_m \sum_{n \neq m} a_{im} a_{in} < X_m h, X_n h >
\end{aligned}$$

La condition du premier ordre pour un extremum de $\ln p(a_i | z_i, h, y_i)$ s'écrit donc :

$$0 = \frac{1}{\sigma_e^2} < X_m h, c_i > - \frac{1}{\sigma_e^2} a_{im} \, ||X_m h||^2 - \frac{1}{\sigma_e^2} \sum_{n \neq m} a_{in} < X_m h, X_n h > + T_{im}$$

avec

$$T_{im} = \begin{cases} \dfrac{\alpha_{1m} - 1}{a_{im}} - \beta_{1m} & \text{si } z_{im} = 1 \\ \dfrac{\alpha_{-1m} - 1}{a_{im}} + \beta_{-1m} & \text{si } z_{im} = -1 \\ -\dfrac{1}{\sigma_{0m}^2} a_{im} & \text{si } z_{im} = 0 \end{cases}$$

et ce quelque soit $m = 1, \ldots, M$.

La condition du premier ordre se reformule donc sous la forme d'une équation $\Phi(a_i) = 0$ avec $\Phi : \mathbb{R}^M \to \mathbb{R}^M$. Cette équation vectorielle non linéaire peut se résoudre numériquement avec la méthode de Newton ; pour appliquer cette méthode nous devons calculer la matrice jacobienne de l'application Φ. Les coefficients d_{mn} de cette matrice s'écrivent :

$$d_{mm} = \frac{\partial \Phi_m}{\partial a_{im}} = -\frac{1}{\sigma_e^2} ||X_m h||^2 + \begin{cases} -\dfrac{\alpha_{1m} - 1}{a_{im}^2} & \text{si } z_{im} = 1 \\ -\dfrac{\alpha_{-1m} - 1}{a_{im}^2} & \text{si } z_{im} = -1 \\ -\dfrac{1}{\sigma_{0m}^2} & \text{si } z_{im} = 0 \end{cases}$$

et, si $m \neq n$:

$$d_{mn} = \frac{\partial \Phi_m}{\partial a_{in}} = -\frac{1}{\sigma_e^2} < X_m h, X_n h >$$

L'itération de Newton est initialisée avec la valeur de (a_i) avant mise à jour.

3.3.4 Mise à jour de z

La mise à jour de z_{im} se fait encore par ICM :

$$z_{im} = \underset{\widetilde{z_{im}} \in \{\pm 1, 0\}}{\arg\max} \ln p\left(\widetilde{z_{im}} | a, (z_{jm})_{j \in \mathcal{N}(i)}\right)$$

soit :

$$z_{im} = \underset{\widetilde{z_{im}} \in \{\pm 1, 0\}}{\arg\max} \left[\ln\left(p(a_{im} | \widetilde{z_{im}})\right) - \beta_1 \sum_{j \in \mathcal{N}(i)} \delta(\widetilde{z_{im}}, z_{jm}) \right] \tag{13}$$

L'argmax est calculé en calculant la fonction objectif pour les trois valeurs possibles de $\widetilde{z_{im}}$ et en conservant la valeur donnant la plus grande valeur à l'objectif.

Nous retrouvons la remarque faite dans [2], partie 3, à savoir que l'équation (13) réalise un compromis entre d'une part l'adaptation aux données (pour le premier terme), et d'autre part, l'homogénéité spatiale de la classification gérée par le second terme. Pour $\beta_1 = 0$ on retrouve une classification sans aucune homogénéité imposée *a priori* ; lorsque $\beta_1 \to +\infty$ on retrouve un vote à la majorité des voisins.

3.3.5 Initialisation

Pour initialiser la classification, nous calculons, pour chaque voxel i et chaque condition expérimentale m, le coefficient de corrélation linéaire entre y_i et $X_m h$. Un voxel dont le décours temporel a une corrélation avec $X_m h$ supérieure à un certain seuil $s > 0$ sera classé comme « activé », un voxel dont la corrélation est inférieure à $-s$ sera classé comme « désactivé », et enfin un voxel avec une corrélation entre $-s$ et s sera classé comme « neutre ». Le seuil s peut être spécifié directement par l'utilisateur, ou calculé automatiquement comme une fraction (définissable par l'utilisateur) du plus grand (resp. du plus petit) coefficient de corrélation, pour chaque condition expérimentale. Nous obtenons également des valeurs initiales pour les niveaux de réponse a_{im} comme suit.

Soit un voxel i et une condition n ; partant de l'équation de base du modèle

$$y_i = \sum_{m=1}^{M} a_{im}(X_m h) + n_i \mathbf{1} + \epsilon_i, \ i = 1, \ldots, J$$

prenons la covariance avec $X_n h$, nous obtenons

$$\mathrm{Cov}(y_i, X_n h) = \sum_m a_{im} \mathrm{Cov}(X_m h, X_n h) + n_i \mathrm{Cov}(X_n h, \mathbf{1}) + \mathrm{Cov}(\epsilon_i, X_n h)$$

et les deux dernières covariances sont nulles. Le vecteur des réponses neuronales

$$A_i = \begin{pmatrix} a_{i1} \\ \vdots \\ a_{iM} \end{pmatrix}$$

vérifie donc l'équation

$$M_i A_i = C_i$$

en notant C_i le vecteur des covariances

$$C_i = \begin{pmatrix} \mathrm{Cov}(y_i, X_1 h) \\ \vdots \\ \mathrm{Cov}(y_i, X_M h) \end{pmatrix}$$

et M_i la matrice de variance-covariance

$$M_i = \begin{pmatrix} \mathrm{Var}(X_1 h) & \mathrm{Cov}(X_1 h, X_2 h) & \cdots & \mathrm{Cov}(X_1 h, X_M h) \\ \mathrm{Cov}(X_2 h, X_1 h) & \mathrm{Var}(X_2 h) & \cdots & \mathrm{Cov}(X_2 h, X_M h) \\ \vdots & \ddots & \ddots & \vdots \\ \mathrm{Cov}(X_M h, X_1 h) & \cdots & \mathrm{Cov}(X_M h, X_{M-1} h) & \mathrm{Var}(X_M h) \end{pmatrix}$$

De même le vecteur des corrélations

$$A'_i = \begin{pmatrix} \mathrm{Corr}(y_i, X_1 h) \\ \vdots \\ \mathrm{Corr}(y_i, X_M h) \end{pmatrix}$$

vérifie

$$M'_i A'_i = C'_i$$

avec

$$C'_i = \begin{pmatrix} \mathrm{Corr}(y_i, X_1 h) \\ \vdots \\ \mathrm{Corr}(y_i, X_M h) \end{pmatrix}$$

et

$$M'_i = \begin{pmatrix} 1 & \mathrm{Corr}(X_1 h, X_2 h) & \cdots & \mathrm{Corr}(X_1 h, X_M h) \\ \mathrm{Corr}(X_2 h, X_1 h) & 1 & \cdots & \mathrm{Corr}(X_2 h, X_M h) \\ \vdots & \ddots & \ddots & \vdots \\ \mathrm{Corr}(X_M h, X_1 h) & \cdots & \mathrm{Corr}(X_M h, X_{M-1} h) & 1 \end{pmatrix}$$

Notons que pour mener les calculs de cette phase, nous initialisons la fonction de réponse hémodynamique à la réponse « canonique »(cf. 2.6).

3.3.6 Algorithme final

Les différentes étapes de mise à jour sont alors enchaînées ainsi :

```
Initialisation (a, z, σ_e, n) ;
Tant que non convergence faire :
        Mise à jour de h ;
        Mise à jour des résidus (σ_e et n) ;
        Mise à jour des z ;
        Si au moins un z a été changé :
                Mise à jour des a ;
                Mise à jour des α_{±1m},β_{±1m} et σ_{0m} ;
```

La convergence est déclarée lorsque, entre deux passages successifs dans la boucle "Tant que", aucun z n'a été modifié et la norme L^2 de la différence entre la fonction h précédente et la fonction h mise à jour est inférieure à un certain seuil.

3.4 Simulation de données

Afin d'évaluer notre approche de traitement d'images, nous devons d'abord simuler des images IRMf réalistes.

La donnée de matrices de design $X_1, \ldots, X_M$ associées à M conditions expérimentales ainsi que d'une HRF h fournit les signaux $X_1h, X_2h, \ldots, X_Mh$. On se donne également deux collections $(\mathcal{A}(1), \ldots, \mathcal{A}(M))$ et $(\mathcal{D}(1), \ldots, \mathcal{D}(M))$ d'ensembles de voxels, l'ensemble $\mathcal{A}(m)$ (resp. $\mathcal{D}(m)$) contenant l'ensemble des voxels activés (resp. désactivés) par la condition m. Les autres voxels sont considérés comme neutres.

Le signal dans le voxel i est donné par

$$y_i = c \sum_{m \in \mathcal{A}(m)} X_m h - c \sum_{m \in \mathcal{D}(m)} X_m h + b\mathbf{1} + \epsilon_i$$

où b est un réel, c un réel positif, et $(\epsilon_i)_i$ est une suite de variables aléatoires vectorielles indépendantes où ϵ_i est un processus AR(1) gaussien ; c'est à dire que quelque soit i,

$$\begin{aligned}
\epsilon_{i,1} &= \sigma e_{i,1} \\
\epsilon_{i,2} &= \rho\epsilon_{i,1} + \sigma e_{i,2} \\
\epsilon_{i,3} &= \rho\epsilon_{i,2} + \sigma e_{i,3} \\
\vdots \quad & \quad \vdots \\
\epsilon_{i,T} &= \rho\epsilon_{i,T-1} + \sigma e_{i,T}
\end{aligned}$$

avec les $(e_{i,t})_{i,t}$ tous indépendants et suivant une loi normale centrée réduite.

Le coefficient $\rho \in [0;1[$ est le coefficient d'autocorrélation du bruit ; dans les applications typiques de l'IRMf sa valeur varie entre 0 et 0.4 (cf. [26], p. 5).

Le coefficient c est fixé à 1000, la valeur de σ (écart-type du bruit) est alors adaptée pour prescrire le rapport signal sur bruit. Ce rapport signal sur bruit est calculé ainsi (cf. [21]) :

$$SNR = \frac{1000\mu}{\sigma}$$

où μ est la moyenne du signal « utile » :

$$\mu = \frac{1}{MT} \sum_{m=1}^{M} ||X_m h||_{\ell^2}.$$

L'expression de σ en fonction de SNR est donc :

$$\sigma = \frac{1000\mu}{SNR}.$$

Les valeurs de SNR sont généralement comprises entre 0.1 et 4.0 (cf. [27]).

3.5 Evaluation de l'approche

3.5.1 Evaluation de l'estimation/détection

Dans un premier temps, nous ne testons que l'estimation/détection des activations ou des désactivations. Nous simulons donc des données à partir de la HRF canonique, et nous forçons le programme à ne pas estimer de HRF mais à utiliser la HRF canonique tout au long du traitement.

Comme paradigme expérimental, nous utilisons deux conditions qui s'alternent toutes les 15 secondes, trois fois chacune. Le TR est fixé à 1 seconde.

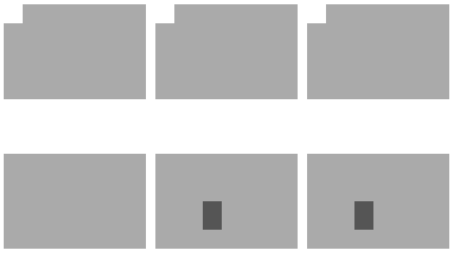
L'image, formée de 6 coupes contigües formant un total de 900 voxels, comporte deux zones parallélépipédiques : une de 12 voxels activés dans la condition 1, désactivés dans la condition 2 ; et une seconde de 8 voxels désactivés dans la condition 1 et activés dans la condition 2. Les images montrées s'intéressent uniquement à la condition 1 (l'autre étant symétrique).

Comme nous pouvons le voir sur la Figure 5, le SNR a bien entendu une grande importance dans la qualité de l'estimation obtenue. La Figure 6 montre que le coefficient de corrélation ρ n'a pas une grande importance sur la reconstruction de l'image.

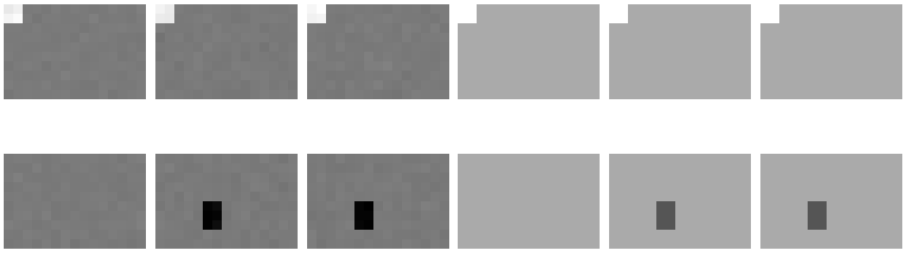
Au niveau du paramètre de régularisation spatiale β_1, nous voyons sur la Figure 7 que plus β_1 est grand, plus notre estimateur recherche des blocs

contigus d'activation ; par contre si β_1 est trop fort, la méthode perd en sensibilité de détection des petites activations. Par exemple, dans l'image (e), (pour $\beta_2 = 5$) le groupe des 8 voxels désactivés n'est plus détecté ; par contre les 12 voxels activés sont toujours bien détectés.

Enfin, la Figure 8 montre que notre méthode de classification par *a priori* markovien donne de biens meilleurs résultats que la technique de seuillage sur les coefficients de corrélation, utilisée pour initialiser l'algorithme d'estimation.

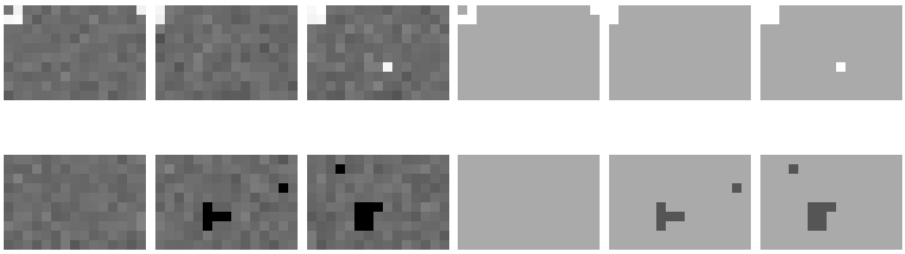


(a) Image originale (non bruitée)



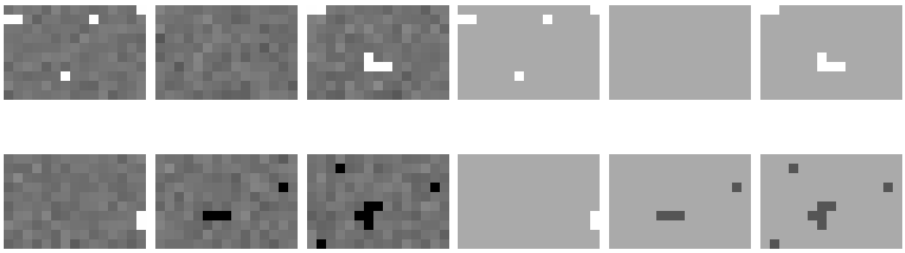
(b) Niveaux de réponses estimés pour SNR=0.1.

(c) Voxels activés estimés pour SNR=0.1.



(d) Niveaux de réponses estimés pour SNR=0.02.

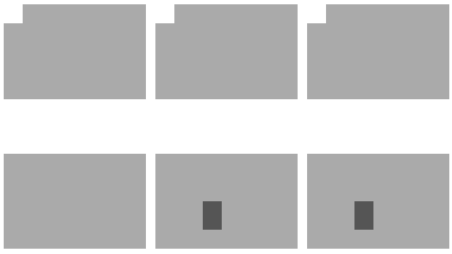
(e) Voxels activés estimés pour SNR=0.02.



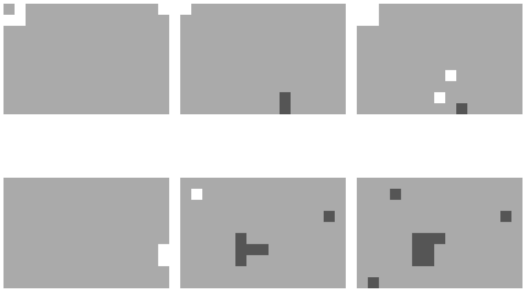
(f) Niveaux de réponses estimés pour SNR=0.01.

(g) Voxels activés estimés pour SNR=0.01.

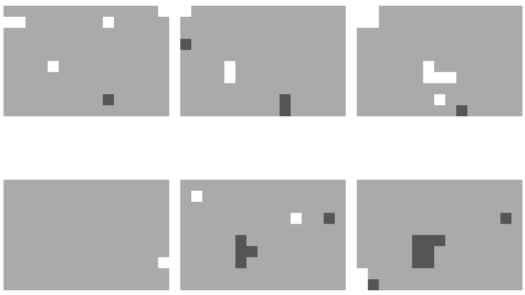
FIG. 5: **Influence du SNR.** ($\beta_1 = 1$, $\rho = 0$).



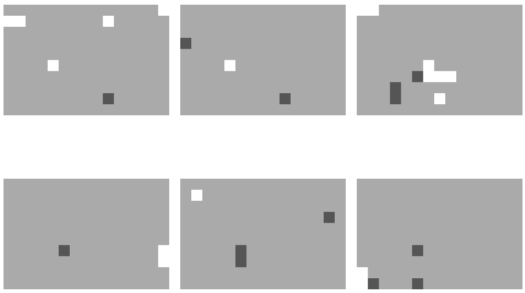
(a) Image originale (non bruitée)



(b) $\rho = 0.2$

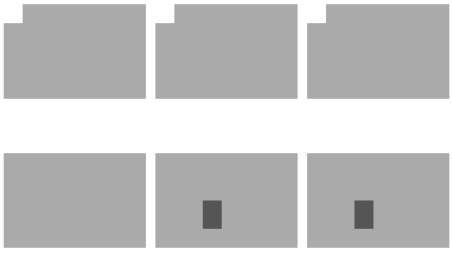


(c) $\rho = 0.4$

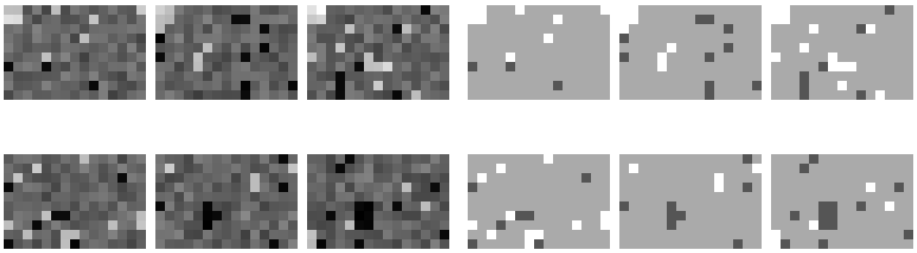


(d) $\rho = 0.6$

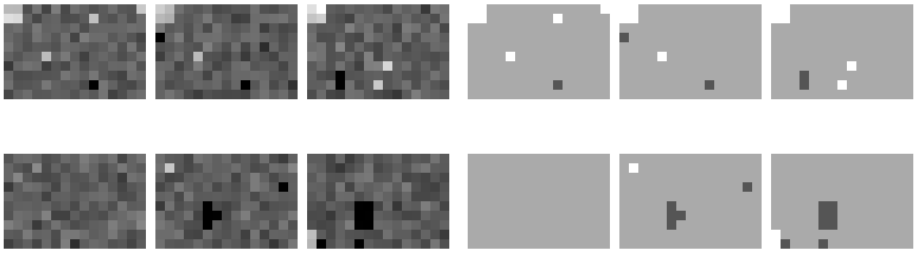
FIG. 6: **Influence du coefficient d'autocorrélation AR(1) ρ. ($\beta_1 = 1$, SNR=0.02)**.



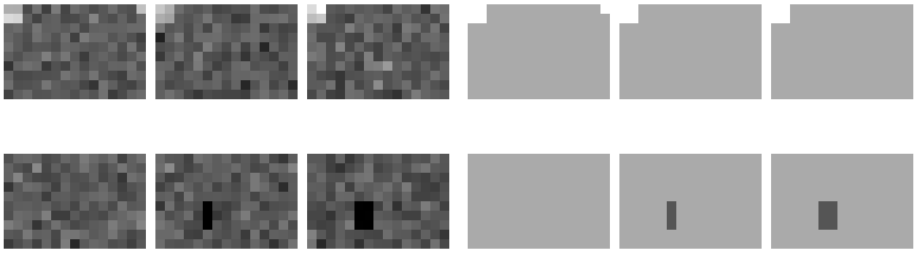
(a) Image originale.



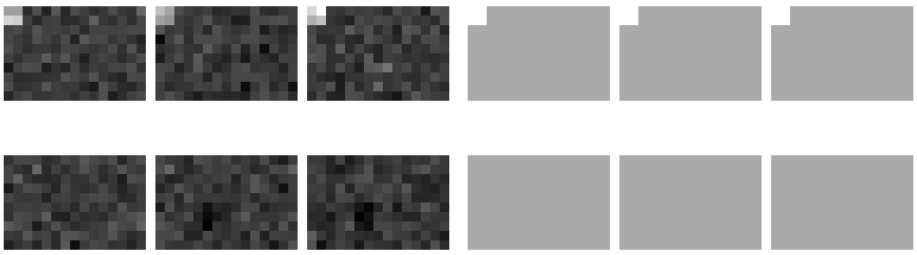
(b) $\beta_1 = 0.5$



(c) $\beta_1 = 1$

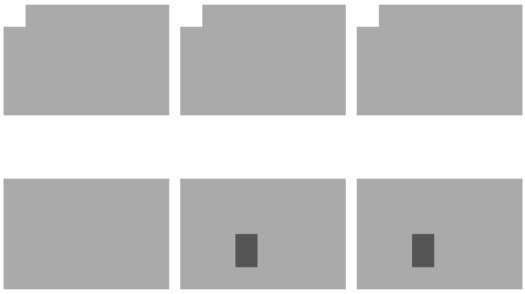


(d) $\beta_1 = 2$

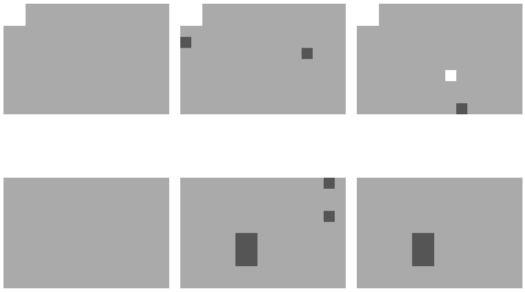


(e) $\beta_1 = 5$

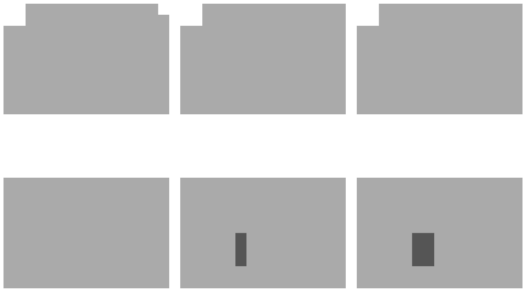
FIG. 7: **Influence du coefficient β_1 (SNR=0.04, ρ=0).** Les niveaux d'activations sont à gauche et les voxels activés à droite.



(a) Image originale



(b) Seuillage sur les coefficients de corrélation (seuil=0.5×le plus grand coefficient de corrélation)



(c) Estimation par ICM ($\beta_1 = 2$)

FIG. 8: **Comparaison des classifications voxels activés/désactivés (SNR=0.04).**

3.5.2 Evaluation de l'estimation de la HRF

Pour tester notre méthode d'estimation de la HRF sur des données faisant sens physiologiquement, nous simulons des données générées à partir de la convolution avec une HRF dont le retour à zéro après la stimulation est anormalement lente ; cette forme de HRF a été observée chez les rats épileptiques (cf. [4], Figure 3 (B)).

Une formule donnant une telle HRF est la suivante (voir Figure 9) :

$$h(t) = \begin{cases} -0.694t^2 + 8.33t & \text{si } t < 6 \\ -10.54 \ln t + 43.88 & \text{si } t \geq 6 \end{cases}$$

Pour tenir compte de cette réponse hémodynamique plus lente, nous choisissons d'estimer la HRF sur une durée de 60 secondes (au lieu de 28 secondes habituellement). Toutes les simulations de cette section ont été effectuées en fixant $\beta_1 = 1$.

SNR élevé Nous voulons d'abord tester le comportement de notre estimation de la HRF pour un niveau de bruit très faible. Pour cela nous choisissons $SNR = 2$. Nous utilisons le même paradigme expérimental qu'à la section précédente, à savoir alternance toutes les 15 secondes de deux conditions.

La HRF estimée est visible en Figure 10.

Ce résultat, peu satisfaisant compte tenu du niveau de bruit très faible, est dû au paradigme expérimental qui ne tient pas compte de la lenteur de la décroissance de la réponse hémodynamique. En effet, si nous changeons le rhythme d'alternance des deux conditions, à savoir que nous alternons trois fois chacune deux conditions toutes les 60 secondes, nous obtenons l'estimation visible en Figure 11.

L'estimation est cette fois-ci très correcte.

SNR faible En conservant le paradigme lent, nous effectuons d'autres simulations en prenant $SNR = 0.03$.

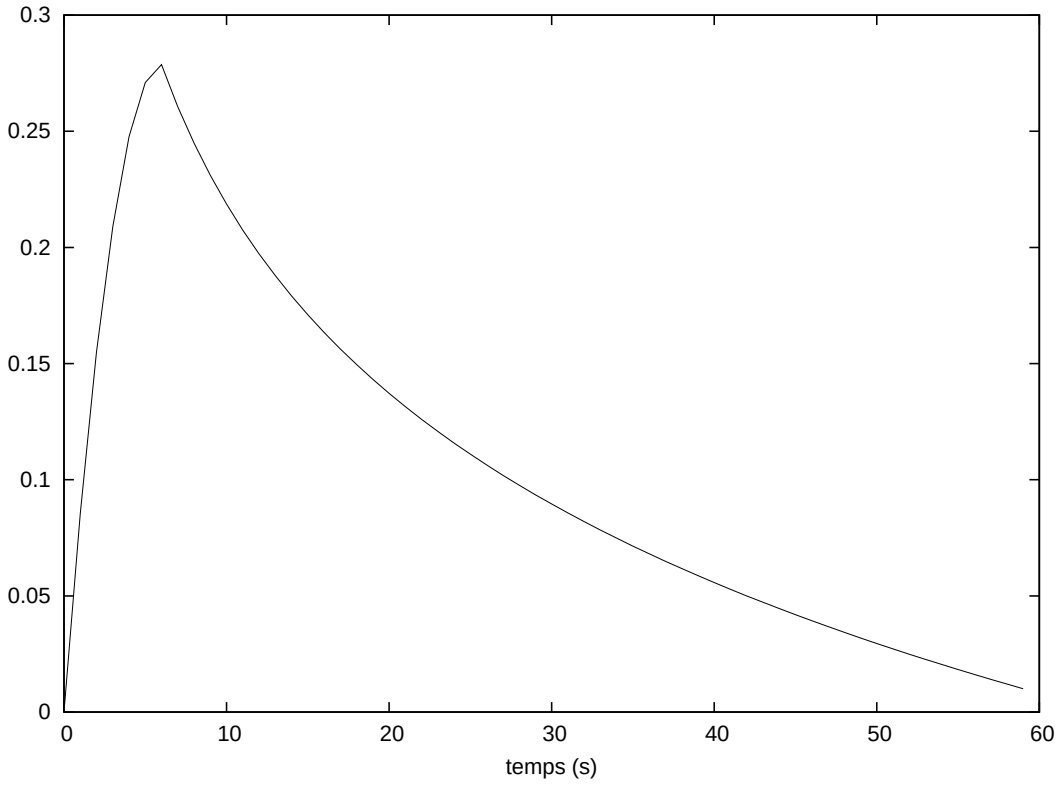


FIG. 9: Profil de la HRF à variation lente utilisée.

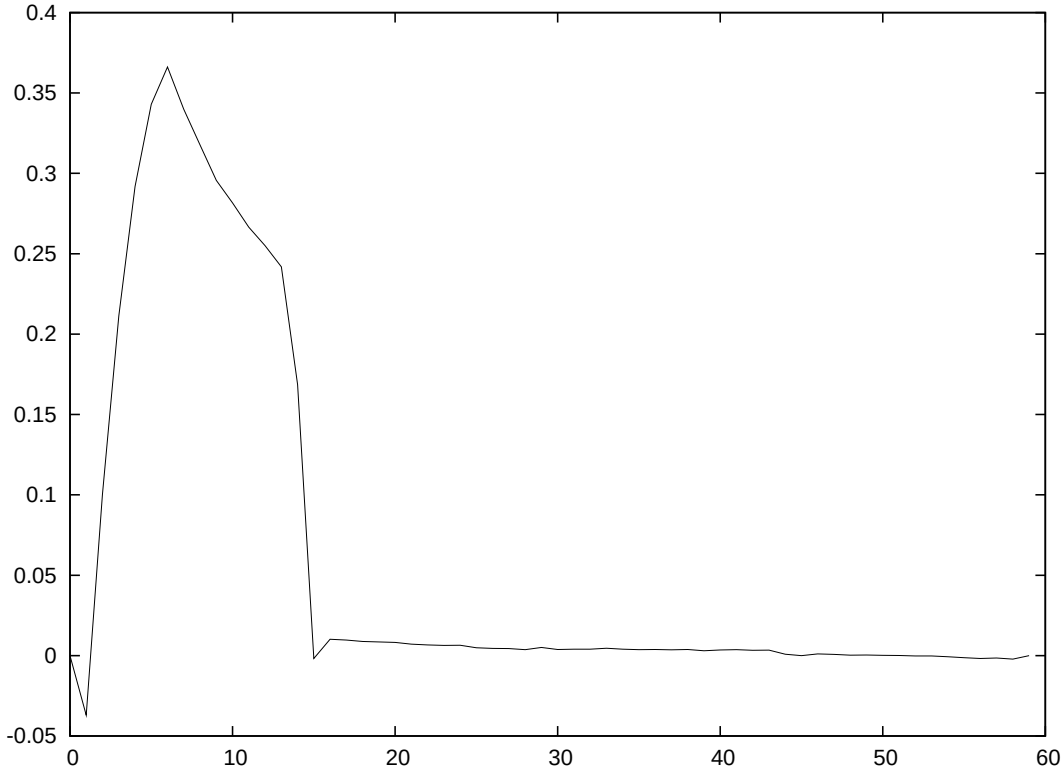


FIG. 10: HRF estimée pour SNR=2, $\beta_2 = 0$, aucun lissage gaussien et un paradigme expérimental « rapide ».

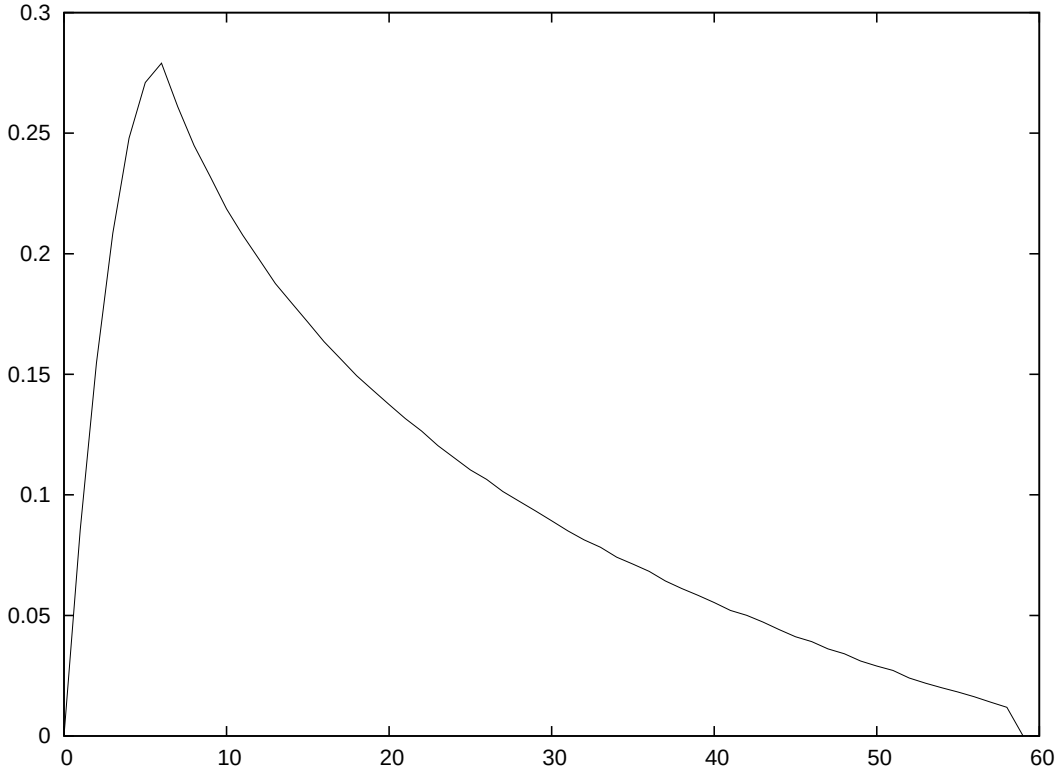
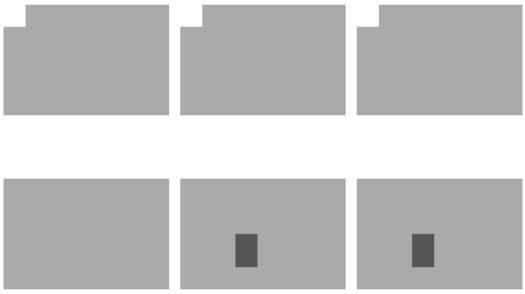
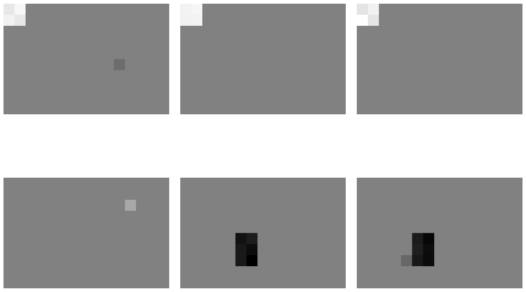


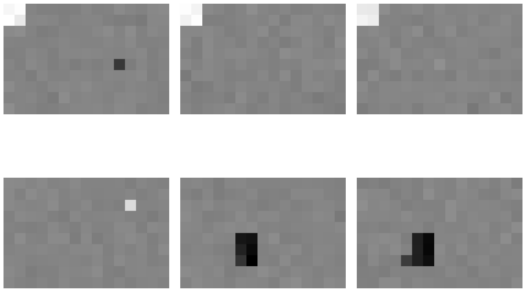
FIG. 11: HRF estimée pour SNR=2, $\beta_2 = 0$, aucun lissage gaussien et un paradigme expérimental « lent ».



(a) Image originale.

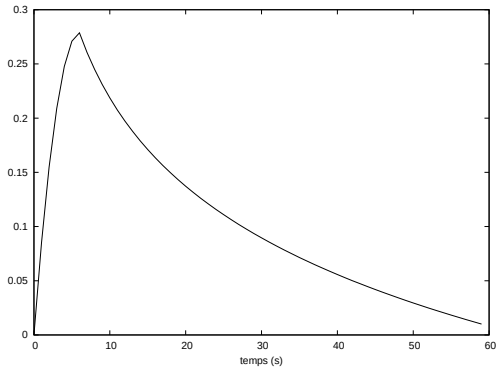


(b) Image reconstruite avec estimation de la HRF.

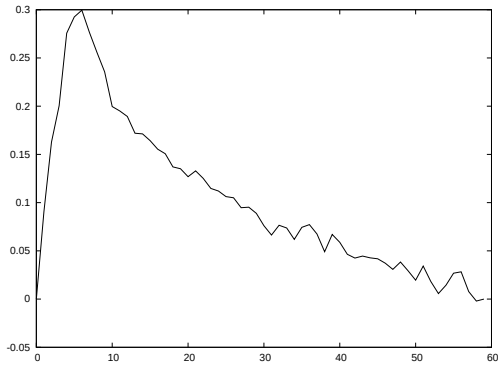


(c) Image reconstruite sans estimation de la HRF.

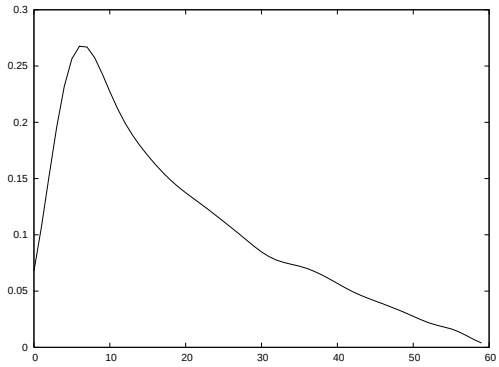
FIG. 12: **Influence de l'estimation de la HRF sur l'estimation des niveaux de réponse.**



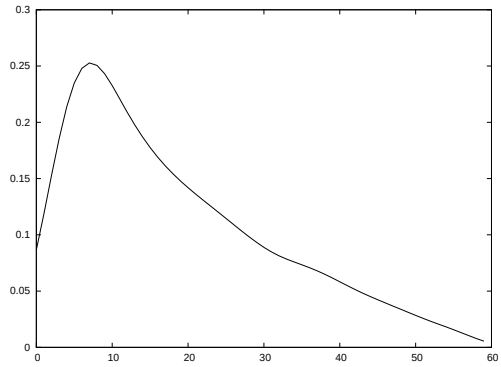
(a) HRF originale.



(b) HRF estimée sans lissage gaussien.

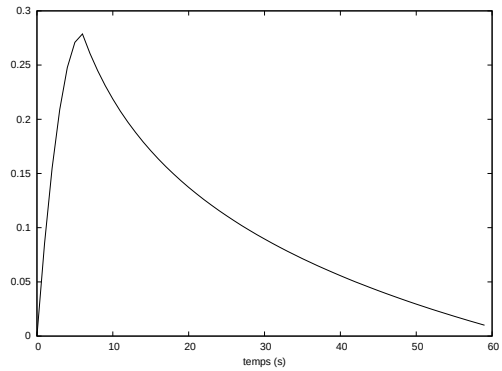


(c) HRF estimée avec lissage gaussien ($BW = 10$)

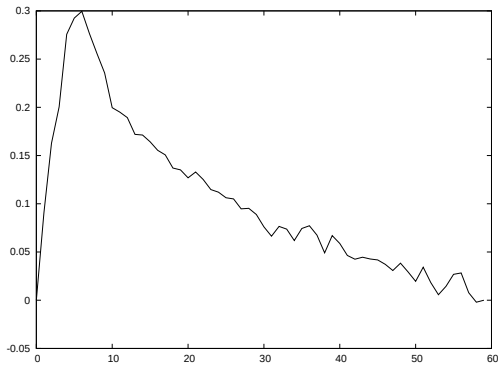


(d) HRF estimée avec lissage gaussien ($BW = 20$)

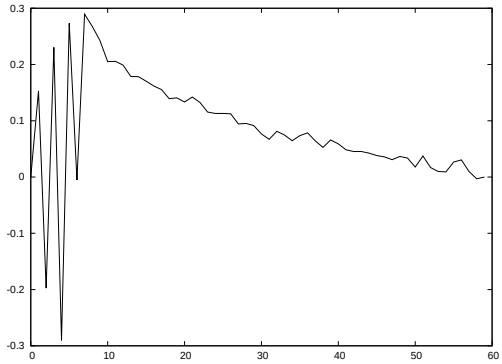
FIG. 13: **Influence du paramètre BW de lissage gaussien ($\beta_2 = 0$).**



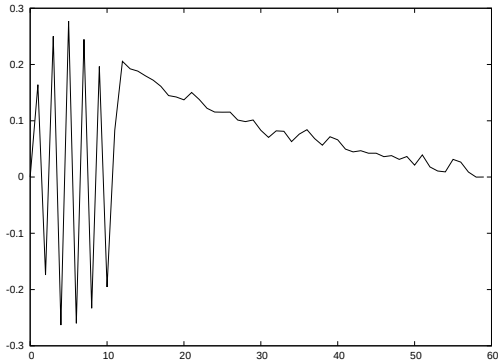
(a) HRF originale.



(b) HRF estimée avec $\beta_2 = 0$.



(c) HRF estimée avec $\beta_2 = 0.002$.



(d) HRF estimée avec $\beta_2 = 0.5$.

FIG. 14: **Influence de β_2 (sans lissage gaussien).**

A la lecture des figures 12 à 14, nous pouvons donc dire que l'estimation de la HRF fournit une meilleure estimation des niveaux de réponse, et que le lissage gaussien donne de bons résultats, à comparer avec le lissage par β_2 qui semble ne pas fonctionner.

3.5.3 Evaluation sur des données réelles

Nous avons testé notre programme sur des données issues d'une expérience de vision des couleurs ; on présente au sujet pendant 200 millisecondes une image en noir et blanc (condition 1), repos durant 12.3 secondes, pendant 200 millisecondes une image en couleur (condition 2), repos durant 12.3 secondes, et encore repos pendant 12.5 secondes (condition 3). Ce bloc de trois conditions est répété 8 fois. Le temps de répétition est de 2.5 secondes. La résolution spatiale est de 3 mm.

Les paramètres utilisés pour le traitement sont visibles dans le fichier de design que nous donnons intégralement en partie A.2.3.

Les données ont été préalablement traitées pour corriger les mouvements éventuels du patient et ont subi un lissage gaussien.

Les résultats, sous forme de contrastes entre les différentes conditions, sont dans les Figures 15 à 17. La HRF estimée est montrée en Figure 18.

L'interprétation de ces résultats n'est pas encore parfaitement terminée, cependant on peut déjà dire que l'on retrouve bien des activations dans les aires visuelles, ce qui est normal.

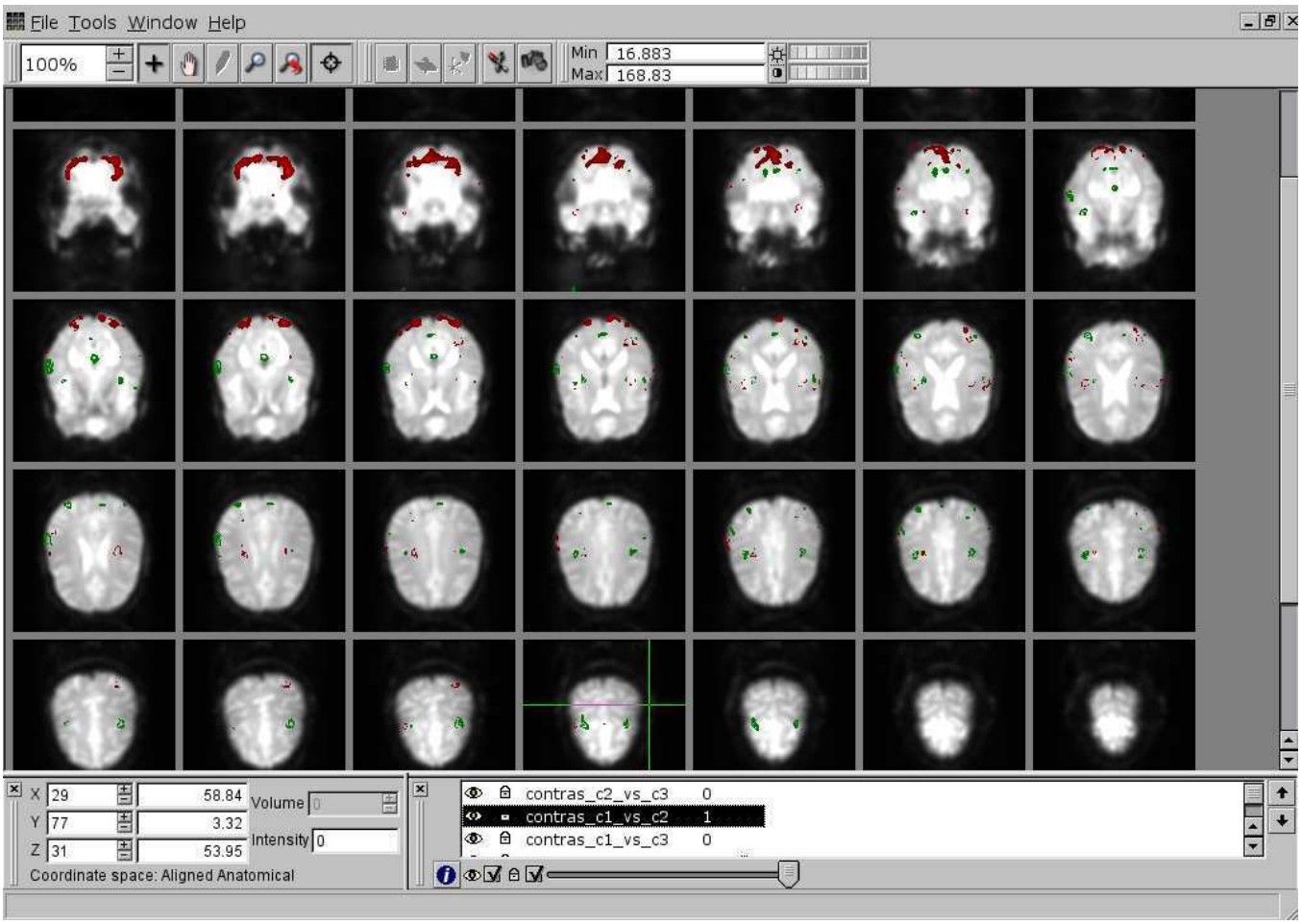


FIG. 15: Condition 1 vs. 2 (en vert les voxels activés dans la condition 1 et non dans la condition 2 ; en rouge les voxels désactivés dans la condition 1 et non dans la condition 2).

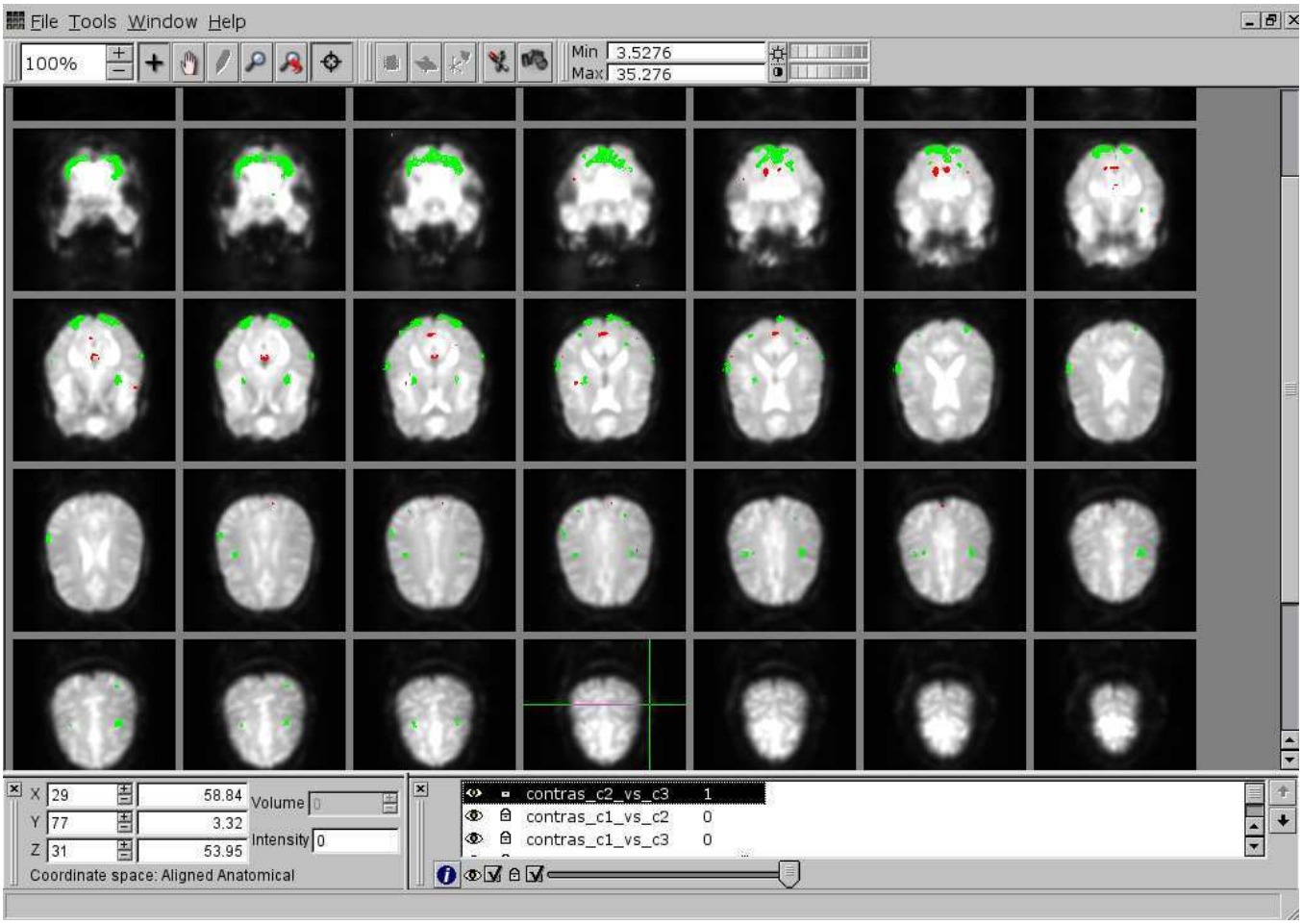


FIG. 16: Condition 2 vs. 3

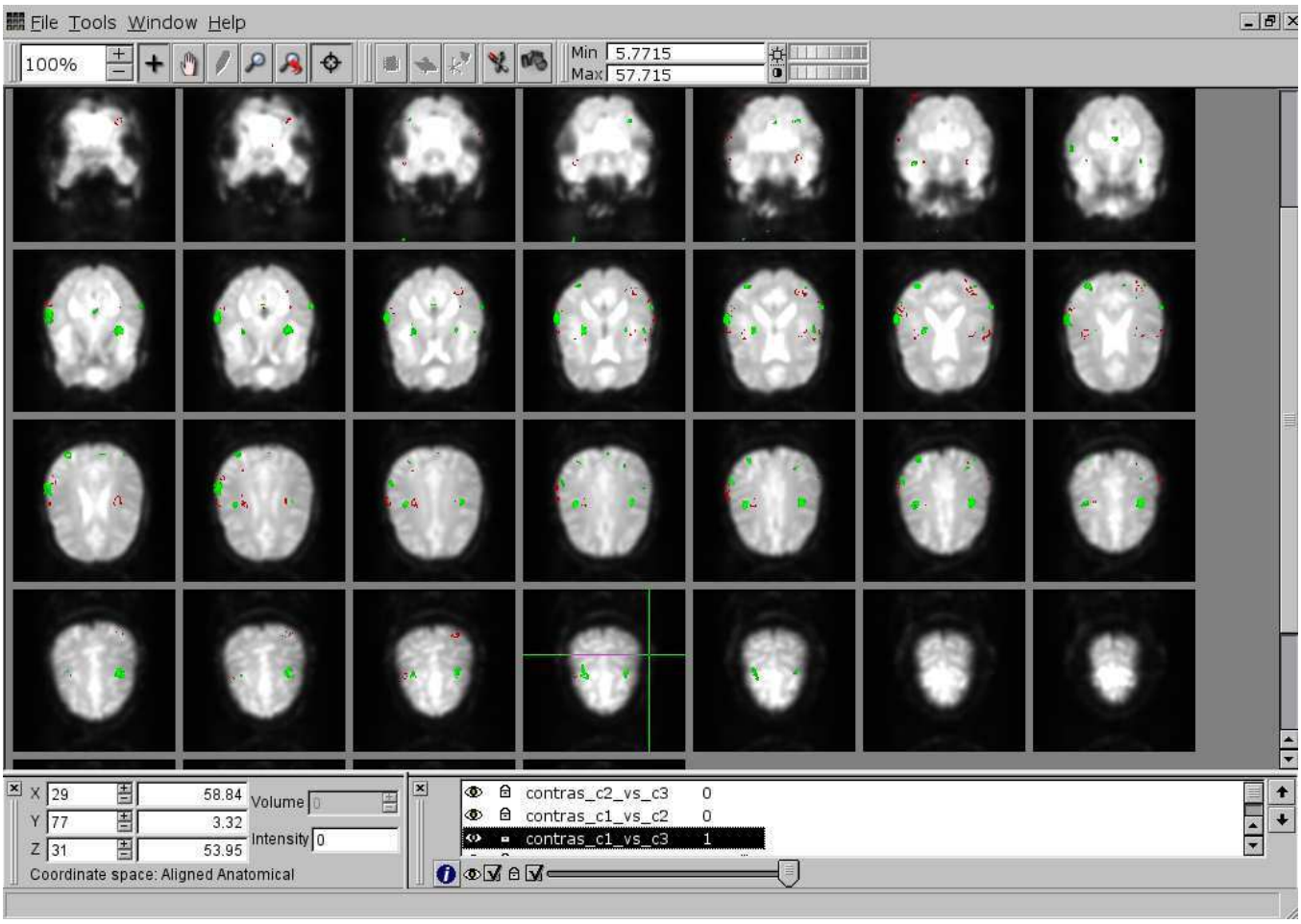


FIG. 17: Condition 1 vs. 3

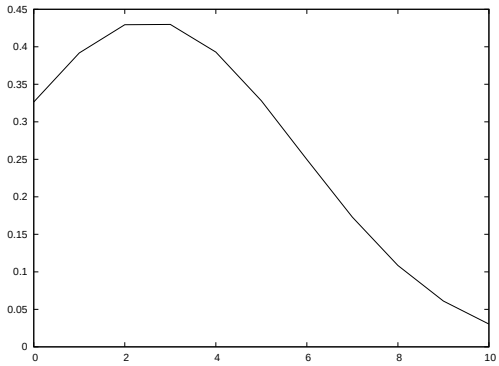


FIG. 18: **Estimation de la HRF (1 unité en abscisses = 2.5 s).**

4 Seconde approche : modèle mélange de gaussiennes et EM variationnel

L'intérêt de notre seconde approche est le remplacement de la méthode ICM par une approche variationnelle d'approximation du posterior, afin de déterminer l'ensemble de la loi a posteriori (et non pas seulement un de ses modes). L'approche variationnelle que nous proposons est équivalente à l'utilisation de l'algorithme EM variationnel. Pour cette approche, nous reprenons le modèle linéaire décrit précédemment, ainsi que la classification des voxels en trois classes. L'interaction spatiale entre les voxels est toujours capturée par un modèle de Potts sur les classes z_{im}.

Pour la modélisation des niveaux de réponse, nous avons considéré un modèle « mélange de trois gaussiennes », i.e.

$$a_{im}|z_{im} \sim \begin{cases} \text{Normale}(\mu_{1m}, \sigma_{1m}^2) & \text{si } z_{im} = 1 \\ \text{Normale}(\mu_{-1m}, \sigma_{-1m}^2) & \text{si } z_{im} = -1 \\ \text{Normale}(0, \sigma_{0m}^2) & \text{si } z_{im} = 0 \end{cases}$$

4.1 Loi a posteriori des z

Nous cherchons à approximer la loi a posteriori des z :

$$p(z|y) = \frac{p(a|z)p(z)}{p(a)} \propto \prod_m \prod_i \left[p(a_{im}|z_{im}) \prod_{j \in \mathcal{N}(i)} \exp(\beta \mathbf{1}_{z_{im}=z_{jm}}) \right]$$

par la loi factorisée suivante :

$$q(z) = \prod_m \prod_i q_{im}(z_{im})$$

avec :

$$q_{im}(z_{im}) = \begin{cases} \lambda_{im} & \text{si} \quad z_{im} = 1 \\ \mu_{im} & \text{si} \quad z_{im} = -1 \\ 1 - \lambda_{im} - \mu_{im} & \text{si} \quad z_{im} = 0 \end{cases}$$

où $(\lambda_{im})_{i,m}$ et $(\mu_{im})_{i,m}$ sont à déterminer de façon à minimiser :

$$KL = \mathbf{E}_q \left[\ln \frac{q(z)}{p(z|a)} \right].$$

On a :

$$\frac{q(z)}{p(z|a)} \propto \prod_{m,i} \left[\frac{q_{im}(z_{im})}{p(a_{im}|z_{im})} \times \prod_{j \in \mathcal{N}(i)} \exp(-\beta \mathbf{1}_{z_{im}=z_{jm}}) \right]$$

soit

$$\ln \frac{q(z)}{p(z|a)} = \text{Cste} + \sum_{m,i} \ln q_{im}(z_{im}) - \ln p(a_{im}|z_{im}) - \beta \sum_{j \in \mathcal{N}(i)} \mathbf{1}_{z_{im}=z_{jm}}$$

Prenons maintenant l'espérance par rapport à q ; nous avons :

$$\mathbf{E}_q \left[\ln q_{im}(z_{im})\right] = \lambda_{im} \ln \lambda_{im} + \mu_{im} \ln \mu_{im} + (1 - \lambda_{im} - \mu_{im}) \times \ln(1 - \lambda_{im} - \mu_{im})$$

et :

$$\begin{aligned} \mathbf{E}_q \left[\mathbf{1}_{z_{im}=z_{jm}}\right] &= \text{proba, sous } q, \text{ que } z_{im} = z_{jm} \\ &= \lambda_{im}\lambda_{jm} + \mu_{im}\mu_{jm} + (1 - \lambda_{im} - \mu_{im})(1 - \lambda_{jm} - \mu_{jm}) \end{aligned}$$

En notant :

$$\ln p(a_{im}|z_{im}) =: \begin{cases} D_1(a_{im}) & \text{si} \quad z_{im} = 1 \\ D_0(a_{im}) & \text{si} \quad z_{im} = 0 \\ D_{-1}(a_{im}) & \text{si} \quad z_{im} = -1 \end{cases}$$

(D_1, D_0 et D_{-1} sont des log-densités de lois normales). on a :

$$\mathbf{E}_q \left[\ln p(a_{im}|z_{im})\right] = \lambda_{im} D_1(a_{im}) + \mu_{im} D_{-1}(a_{im}) + (1 - \lambda_{im} - \mu_{im}) D_0(a_{im})$$

Donc :

$$\begin{aligned} KL &= \text{Cste} + \sum_{m,i} [\lambda_{im} \ln \lambda_{im} + \mu_{im} \ln \mu_{im} + (1 - \lambda_{im} - \mu_{im}) \times \ln(1 - \lambda_{im} - \mu_{im}) \\ &\quad -\lambda_{im} D_1(a_{im}) - \mu_{im} D_{-1}(a_{im}) - (1 - \lambda_{im} - \mu_{im}) D_0(a_{im}) \\ &\quad -\beta \times \sum_{j \in \mathcal{N}(i)} \lambda_{im}\lambda_{jm} + \mu_{im}\mu_{jm} + (1 - \lambda_{im} - \mu_{im})(1 - \lambda_{jm} - \mu_{jm})] \end{aligned}$$

D'où en dérivant par rapport à λ_{im} :

$$0 = \ln \lambda_{im} - \ln(1 - \lambda_{im} - \mu_{im}) - D_1(a_{im}) + D_0(a_{im}) - 2\beta \sum_{j \in \mathcal{N}(i)} (2\lambda_{jm} + \mu_{jm} - 1) \tag{14}$$

et

$$0 = \ln \mu_{im} - \ln(1 - \lambda_{im} - \mu_{im}) - D_{-1}(a_{im}) + D_0(a_{im}) - 2\beta \sum_{j \in \mathcal{N}(i)} (2\mu_{jm} + \lambda_{jm} - 1) \tag{15}$$

Ces relations – appelées équations de champ moyen – forment, pour chaque $m = 1, \ldots, M$, un système de $2 \times J$ équations non linéaires, dont les inconnues sont les λ_{im} et les μ_{im}.

La méthode de Newton ne fonctionne pas ici, car au cours des itérations de Newton, les valeurs « courantes » des λ_{im} et des μ_{im} peuvent sortir de l'intervalle $[0;1]$, ce qui fait que l'itération n'est plus bien définie. Nous devons donc résoudre le système par point fixe.

Pour cela, cherchons à exprimer λ_{im} et μ_{im} en fonction de λ_{jm} et μ_{jm} (pour $j \in \mathcal{N}(i)$). Posons $\lambda = \lambda_{im}$ et $\mu = \mu_{im}$; les relations (14) et (15) se réécrivent :

$$(S) \left\{ \begin{array}{rcl} k & = & \ln \dfrac{\lambda}{1-(\lambda+\mu)} \\ l & = & \ln \dfrac{\mu}{1-(\lambda+\mu)} \end{array} \right.$$

avec

$$k = D_1(a_{im}) - D_0(a_{im}) + 2\beta \sum_{j \in \mathcal{N}(i)} (2\lambda_{jm} + \mu_{jm} - 1)$$

$$l = D_{-1}(a_{im}) - D_0(a_{im}) + 2\beta \sum_{j \in \mathcal{N}(i)} (2\mu_{jm} + \lambda jm - 1)$$

Posons $K = e^k$ et $L = e^l$. Le système (S) est équivalent à

$$\left\{ \begin{array}{rcl} K & = & \dfrac{\lambda}{1-(\lambda+\mu)} \\ L & = & \dfrac{\mu}{1-(\lambda+\mu)} \end{array} \right.$$

Soit :

$$\lambda = \frac{K}{1+K+L} = \frac{e^k}{1+e^k+e^l}$$

et

$$\mu = \frac{L}{1+K+L} = \frac{e^l}{1+e^k+e^l}$$

Pour des raisons de stabilité numérique, nous devons remplacer ces formules par des formules équivalentes : si $k \geq l$ on utilise :

$$\lambda_{im} = \frac{1}{e^{-k}+1+e^{l-k}} \tag{16}$$

$$\mu_{im} = \frac{e^{l-k}}{e^{-k}+1+e^{l-k}} \tag{17}$$

et si $k < l$:

$$\lambda_{im} = \frac{e^{k-l}}{e^{-l}+1+e^{k-l}} \tag{18}$$

$$\mu_{im} = \frac{1}{e^{-l}+1+e^{k-l}} \tag{19}$$

La résolution des équations (14) et (15) se fait donc itérativement, en appliquant (16)-(19). L'initialisation des itérations se fait en résolvant les équations découplées (14)-(15) écrites pour $\beta = 0$:

$$\lambda = \frac{\mathbf{D}_1}{\mathbf{D}_1 + \mathbf{D}_0 + \mathbf{D}_{-1}}$$

$$\mu = \frac{\mathbf{D}_{-1}}{\mathbf{D}_1 + \mathbf{D}_0 + \mathbf{D}_{-1}}$$

en posant $\mathbf{D}_c = \exp D_c(a_{im}) = p(a_{im}|z_{im} = c)$ pour $c = -1, 0, 1$.

4.2 Loi a posteriori des a

La loi a posteriori des niveaux de réponse $a = (a_{im})_{i,m}$ est approchée par :

$$q_a(a) = p(y|a) \prod_{i,m} p(a_{im}|z_{im} = 1)^{q_{im}(z_{im}=1)} p(a_{im}|z_{im} = -1)^{q_{im}(z_{im}=-1)} p(a_{im}|z_{im} = 0)^{q_{im}(z_{im}=0)}$$

D'où en remplacant $p(y|a)$ par sa valeur et en passant au logarithme :

$$\ln q_a(a) = \sum_i \left[\left(-\frac{||c_i - \sum_m a_{im} X_m h||^2}{2\sigma_e^2} \right) + \sum_m \sum_{c \in \{-1,0,1\}} q_{im}(z_{im} = c) \ln p(a_{im}|z_{im} = c) \right]$$

Soit, d'après le calcul préliminaire effectué en 3.3.3,

$$\begin{aligned} \ln q_a(a) &= \text{Cste} + \sum_i \sum_m \Big[\frac{2a_{im} < X_m h, c_i >}{2\sigma_e^2} - \frac{1}{2} \Big(\frac{a_{im}^2 ||X_m h||^2}{\sigma_e^2} \\ &\quad + \sum_{n \neq m} a_{im} a_{in} \frac{< X_m h, X_n h >}{\sigma_e^2} + \sum_{c \in \{-1,0,1\}} \frac{q_{im}(c)}{\sigma_{cm}^2} (a_{im} - \mu_c)^2 \Big) \Big] \end{aligned}$$

Par conséquent, sous q_a, les vecteurs $(a_{i\cdot})_i$ sont des vecteurs gaussiens indépendants ; le vecteur $a_{i\cdot}$ suit une loi normale dont la moyenne $\widehat{a}_i$ et la matrice de covariance Σ_i sont trouvés en identifiant les termes apparaissant dans l'expression de $\ln q_a(a)$ ci-dessus. En effet, si X suit une loi Normale(μ, Σ) on a :

$$\ln p(x) = \text{Cste} - \frac{1}{2} x^T \Sigma^{-1} x + \mu^T \Sigma^{-1} x.$$

En identifiant les termes linéaires et quadratiques on trouve donc :

$$\left(\Sigma^{-1}\right)_{mn} = \begin{cases} \dfrac{||X_m h||^2}{\sigma_e^2} + \displaystyle\sum_{c \in \{-1,0,1\}} \dfrac{q_{im}(z_{im} = c)}{\sigma_{cm}^2} & si \quad n = m \\ \dfrac{< X_m h, X_n h >}{\sigma_e^2} & si \quad n \neq m \end{cases}$$

$$\left(\Sigma^{-1}\widehat{a}_i\right) = \frac{< X_m h, c_i >}{\sigma_e^2} + q_{im}(z_{im} = 1)\frac{\mu_1}{\sigma_1^2} + q_{im}(z_{im} = -1)\frac{\mu_{-1}}{\sigma_{-1}^2}$$

Le calcul de la loi a posteriori des a se ramène donc à l'inversion de J matrices $M \times M$.

4.3 Comparaison de conditions

4.3.1 Contraste

Etant donné un vecteur $k \in \mathbf{R}^M$ de contrastes, calculons la probabilité (a posteriori) que le contraste $< k, a_i >$ d'un voxel soit positif :

$$P(< k, a_i > \geq 0) = \frac{1}{\sqrt{2\pi s^2}} \int_0^\infty e^{-\frac{(x-m)^2}{2s^2}} \mathrm{d}x$$

En effet, a posteriori, $< k, a_i >$ est une v.a. réelle gaussienne de moyenne $m =< k, \widehat{a}_i >$ et de variance $s^2 = k^T \Sigma_i k$.

Après changement de variable, cette probabilité s'exprime à partir de la fonction d'erreur erf :

$$P(< k, a_i > \geq 0) = \frac{1}{2}\left(1 - \mathrm{erf}\left(-\frac{m}{\sqrt{2}s}\right)\right)$$

où

$$\mathrm{erf}(x) = \int_0^x e^{-t^2} \mathrm{d}t.$$

4.3.2 Divergence KL

Pour comparer les conditions m et n, nous pouvons également utiliser la divergence de Kullback-Leibler entre les deux lois a posteriori de a_{im} et a_{in} pour chaque voxel i, à l'instar de [15].

Ces deux lois sont gaussiennes, nous utilisons donc les formules suivantes de calcul de la divergence de KL entre deux gaussiennes, trouvées dans [9] :

$$KL = \frac{1}{2}\ln\frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2}{\sigma_2^2} - 1 + \frac{(\mu_1 - \mu_2)^2}{\sigma_2^2}$$

pour calculer la divergence KL entre deux gaussiennes $\mathcal{N}(\mu_1, \sigma_1^2)$ et $\mathcal{N}(\mu_2, \sigma_2^2)$.

4.4 Estimation de h

Nous donnons une estimation ponctuelle (maximum de vraisemblance) du vecteur de HRF h, choisie de manière à maximiser :

$$\mathbf{E}_{q_a}\left[\ln p(y|a, h)\right].$$

La maximisation de cette quantité est justifiée par le cadre EM variationnel.

On a :

$$p(y|a,h) = \prod_i p(y_i|a_i,h)$$

et :

$$\ln p(y_i|a_i) = -\frac{1}{2\sigma_e^2}\left|\left|c_i - \sum_m a_{im}X_m h\right|\right|^2$$

soit :

$$\begin{aligned}\ln p(y_i|a_i) = \quad & \text{Cste} + \frac{1}{\sigma_e^2}\sum_m a_{im} < X_m h, c_i > -\frac{1}{2\sigma_e^2}\sum_m a_{im}^2 ||X_m h||^2 \\ & -\frac{1}{2\sigma_e^2}\sum_m \sum_{n\neq m} a_{im}a_{in} < X_m h, X_n h >\end{aligned}$$

En utilisant les formules :

$$\mathbf{E}(X^2) = \text{Var}(X) + \mathbf{E}(X)^2$$

et

$$\mathbf{E}(XY) = \text{Cov}(X,Y) + \mathbf{E}(X)\mathbf{E}(Y)$$

Nous trouvons comme quantité à maximiser en h :

$$\begin{aligned}\mathbf{E}_{q_a}\left[\ln p(y|a,h)\right] = \quad & \text{Cste} + \sum_i \Big\{\frac{1}{\sigma_e^2}\sum_m \widehat{a}_{im} < X_m h, c_i > -\frac{1}{2\sigma_e^2}\sum_m (\widehat{a}_{im}^2 + \Sigma_{i,mm})||X_m h||^2 \\ & -\frac{1}{2\sigma_e^2}\sum_m \sum_{n\neq m} (\widehat{a}_{im}\widehat{a}_{in} + \Sigma_{i,mn}) < X_m h, X_n h > \Big\}\end{aligned}$$

où $\Sigma_{i,mn}$ est le coefficient (m,n) de la matrice Σ_i, et $\widehat{a}_{im}$ est la m-ème composante du vecteur $\widehat{a}_i$.

La quantité à maximiser est donc similaire à celle de 3.3.2, en remplaçant a_{im}, a_{im}^2 et $a_{im}a_{in}$ par leurs équivalents tildés :

$$\widetilde{a_{im}} = \widehat{a}_{im}$$

$$\widetilde{a_{im}^2} = \widehat{a}_{im}^2 + \Sigma_{i,mm}$$

$$\widetilde{a_{im}a_{in}} = \widehat{a}_{im}\widehat{a}_{in} + \Sigma_{i,mn}$$

4.5 Estimation de β_1

Nous pouvons estimer le coefficient β_1 du modèle de Potts de la manière suivante : notons N le nombre de cliques homogènes :

$$N = \sum_{i,m} \sum_{j \in \mathcal{N}(i)} \mathbf{1}_{z_{im}=z_{jm}}$$

On peut déterminer la valeur de N à partir de la classification initiale (3.3.5).

Par ailleurs, pour chaque valeur de β_1, on peut calculer la loi $q(z)$ approchant celle de z sachant$y$, puis calculer l'espérance de N sous cette loi $q(z)$ avec la formule :

$$\mathbf{E}(N) = \sum_{i,m} \sum_{j \in \mathcal{N}(i)} q(z_{im}=1)q(z_{jm}=1) + q(z_{im}=0)q(z_{jm}=0) + q(z_{im}=-1)q(z_{jm}=-1)$$

Notre estimation de β_1 – justifiée par la forme de l'estimateur du maximum de vraisemblance de β pour un modèle de Potts (cf. [16]) – consiste à choisir β_1 pour que la valeur observée de N et son espérance soient égales.

En pratique, nous utilisons une recherche dichotomique pour trouver une valeur de β_1 satisfaisante. Cette méthode est justifiée par le fait que la fonction $\beta \mapsto \mathbf{E}(N)$ est croissante.

5 Troisième approche : modèle auto-gaussien

Dans cette approche, nous abandonnons la classification des voxels, et choisissons un prior auto-gaussien sur les a :

$$p(a) \propto \prod_{i,m} \exp\left(-\beta_3 \sum_{j \in \mathcal{N}(i)} (a_{im} - a_{jm})^2\right)$$

tout en conservant par ailleurs le même modèle que dans la deuxième approche.

On trouve :

$$p(a|y) \propto p(y|a)p(a) \propto \prod_i \exp\left[-\frac{||c_i - \sum_m a_{im} X_m h||^2}{2\sigma_e^2} \times \prod_m \prod_i \exp(-\beta_3 \sum_{j \in \mathcal{N}(i)} (a_{im} - a_{jm})^2\right]$$

D'où

$$\begin{aligned} \ln p(a|y) &= \text{Cste} + \sum_i \frac{-1}{2\sigma_e^2}\Big[-2\sum_m a_{im} < X_m h, c_i > + \sum_m a_{im}^2 ||X_m h||^2 \\ &\quad + \sum_m \sum_{n \neq m} a_{im} a_{in} < X_m h, X_n h > \Big] + \sum_m \sum_i -\beta_3 \sum_{j \in \mathcal{N}(i)} (a_{im} - a_{jm})^2 \end{aligned}$$

soit encore :

$$\begin{aligned}
\ln p(a|y) &= \text{Cste} + \sum_i \sum_m \frac{< X_m h, c_i >}{\sigma_e^2} a_{im} \\
&+ \sum_i \sum_m \left(-\frac{||X_m h||^2}{2\sigma_e^2} - (1 + |\mathcal{N}(i)|)\beta_3 \right) a_{im}^2 \\
&+ \sum_i \sum_{j \in \mathcal{N}(i)} \sum_m \left(-\frac{< X_m h, X_n h >}{2\sigma_e^2} + 2\beta_3 \right) a_{im} a_{in}
\end{aligned}$$

La loi a posteriori des a est donc une loi normale, dont les paramètres peuvent être trouvés comme précédemment en identifiant les termes linéaires et quadratiques dans l'expression au-dessus.

6 Conclusion et perspectives

6.1 Conclusion

Nous avons donc présenté des méthodes automatisées de dépouillement des données issues d'une expérience d'IRM fonctionnelle. Les apports de ces méthodes par rapport aux modèles courants sont l'estimation non-paramétrique de la HRF et la prise en compte de l'homogénéité spatiale des données à traiter dans un cadre bayésien bien établi, et l'utilisation d'algorithmes d'inférence rapides.

Voyons maintenant quelques uns des axes d'amélioration.

6.2 Extensions possibles

6.2.1 Meilleure prise en compte du bruit

Un modèle de bruit AR(1) pourrait être estimé dans l'équation (7) pour tenir compte de la corrélation temporelle des erreurs (cf. [26]).

Nous avons commencé à implémenter la gestion du bruit AR(1).

6.2.2 Estimation de la HRF par zones

A l'aide de logiciels de parcellisation, on peut délimiter des zones d'intérêt dans le cerveau qui sont anatomiquement ou fonctionnellement homogènes. Par exemple, dans le cadre d'une expérience sur la vision humaine, on peut délimiter les différentes aires visuelles V1, V2, etc. L'idée est d'estimer une HRF dans chacune de ces régions, et non plus une HRF globale pour tout le cerveau.

Plus précisément, on cherche à estimer $h = (h_1, \ldots, h_k)$ (où k est le nombre de régions, et h_r la HRF de la région r) ; on contraint les HRF de deux régions voisines à être « proches » en rajoutant un champ de Markov auto-gaussien comme loi *a priori* sur h :

$$p(h) \propto \exp\left(-\beta_3 \sum_{r=1}^{k} \sum_{s \in \mathcal{N}(r)} ||h_r - h_s||^2\right)$$

L'équation (7) est alors remplacée par

$$y_i = \sum_{m=1}^{M} a_{im}(X_m h_{r(i)}) + n_i \mathbf{1} + \epsilon_i, \; i = 1, \ldots, J$$

où $r(i)$ désigne la région à laquelle appartient le voxel i.

On peut encore améliorer ce modèle, en considérant qu'un voxel n'est jamais exactement dans une région ou une autre, mais que la délimitation des zones nous apporte une connaissance floue, exprimable sous forme de probabilités d'appartenance d'un voxel à chacune des régions. La HRF propre à un voxel est alors une combinaison des différentes HRF régionales estimées, chacune apparaissant avec un poids d'autant plus important que la probabilité d'appartenance du voxel à la zone est élevée. Si la différence entre les HRFs estimées est suffisament importante, on peut même imaginer qu'une telle estimation permettra de raffiner la délimitation des zones, deux voxels avec une HRF « proche » ayant une probabilité plus importante d'être dans une même zone.

Références

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 32(2) :192–236, 1974.

[2] J. Besag. On the statistical analysis of dirty images. *Journal of Royal Statistics Society*, 48 :259–302, 1986.

[3] J. T. Chang. Markov Random Fields and Hidden Markov Models, 1999. `http://www.cse.ucsc.edu/~chang/203/mrf_short.ps`.

[4] O. David, I. Guillemain, S. Saillet, S. Reyt, C. Deransart, C. Segebarth, and A. Depaulis. Identifying neural drivers with functional MRI : an electrophysiological validation. *PLoS Biology*, 6(12) :2683–97, 2008.

[5] Wellcome Trust Centre for Neuroimaging. Statistical Parametric Mapping. `http://www.fil.ion.ucl.ac.uk/spm`.

[6] R.S.J. Frackowiak, K.J. Friston, R.J. Dolan, C.D. Frith, C.J. Price, J.T. Ashburner, S. Zeki, and W.D. Penny. *Human brain function*. Academic Press, 2004. Disponible à l'adresse `http://www.fil.ion.ucl.ac.uk/spm/doc/books/hbf2/`.

[7] A. Gelman. *Bayesian data analysis*. CRC press, 2004.

[8] D.A. Handwerker, J.M. Ollinger, and M. D'Esposito. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *Neuroimage*, 21(4) :1639–1651, 2004.

[9] JR Hershey, PA Olsen, and I.B.M.T.J.W.R. Center. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 4, 2007.

[10] N. Johnson, S. Kotz, and N. Balakrishnan. Continuous Univariate Distributions, vol. 1-2, 1994.

[11] M.I. Jordan. *Learning in graphical models*. Kluwer Academic Publishers, 1998.

[12] LAPACK++ a library for high performance linear algebra computations. `http://lapackpp.sourceforge.net`.

[13] N.A. Lazar. *The statistical analysis of functional MRI data*. Springer Verlag, 2008.

[14] S. Makni. *Estimation and detection of brain activity in functional magnetic resonance imaging taking account anatomical information*. PhD thesis, Université Paris XI, 2006.

[15] 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.

[16] R. Morris, X. Descombes, and J. Zerubia. An analysis of some models used in image segmentation. *Rapport de recherche INRIA*, 1996.

[17] Neuroimaging Informatics Technology Initiative. `http://nifti.nimh.nih.gov/`.

[18] W. Penny, S. Kiebel, and K. Friston. Variational Bayesian inference for fMRI time series. *NeuroImage*, 19(3) :727–741, 2003.

[19] J. Polzehl and V. Spokoiny. Propagation-separation approach for local likelihood estimation. *Probability Theory and Related Fields*, 135(3) :335–362, 2006.

[20] K. Tabelow, J. Polzehl, H.U. Voss, and V. Spokoiny. Analyzing fMRI experiments with structural adaptive smoothing procedures. *NeuroImage*, 33(1) :55–62, 2006.

[21] C. Triantafyllou, R.D. Hoge, and L.L. Wald. Effect of spatial smoothing on physiological noise in high-resolution fMRI. *Neuroimage*, 32(2) :551–557, 2006.

[22] DG Tzikas, AC Likas, and NP Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6) :131–146, 2008.

[23] T. Vincent, P. Ciuciu, and J. Idier. Spatial Mixture Modelling for the Joint Detection-Estimation of Brain Activity in fMRI. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 1, 2007.

[24] M.W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S.M. Smith. Bayesian analysis of neuroimaging data in FSL. *Neuroimage*, 2008.

[25] M.W. Woolrich, M. Jenkinson, J.M. Brady, and S.M. Smith. Fully Bayesian spatio-temporal modeling of fMRI data. *IEEE transactions on medical imaging*, 23(2) :213–231, 2004.

[26] KJ Worsley, CH Liao, J. Aston, V. Petre, GH Duncan, F. Morales, and AC Evans. A general statistical analysis for fMRI data. *NeuroImage*, 11(5 ; PART 2) :648–648, 2000.

[27] Signal and noise in fMRI, 2003. `http ://www.biac.duke.edu/education/courses/fall03/fmri/`.

A Première implémentation

A.1 Description

Nous avons implémenté en C++, à l'aide de la librairie LAPACK++ [12], la première approche (Gamma-Gamma-Gaussien et ICM) décrite dans ce mémoire. Les deux programmes principaux sont :

- `analyse`, qui effectue le traitement d'une série d'images au format Nifti ;
- `simul`, qui permet d'effectuer les simulations.

Nous avons également écrit deux programmes simples : l'un pour effectuer la conversion d'images Nifti 3D vers PGM (pour traitement ultérieur des images) et un autre pour calculer la différence entre deux images Nifti.

A.2 Format de fichier de design

Le fichier de design est un fichier texte contenant des *paramètres* et la description des conditions ainsi que leurs instants d'arrivées (*onsets*).

A.2.1 Paramètres

- **fichier** : le nom du fichier à partir duquel lire les données 4D ; si ce paramètre se termine par un underscore (_) une série d'images 3D est lue ;
- **image1** : si une série d'images 3D est lue, numéro de la première image à traiter (défaut 1) ;
- **tr** : temps de répétition en secondes ;
- **duree** : durée de l'expérience en secondes ;
- **hrfsample** : période d'échantillonnage de la HRF en secondes (défaut=tr) ;
- **hrfcano** : si présent, la HRF n'est pas estimée (défaut : la HRF est estimée) ;
- **ar1** : si présent, un modèle AR(1) est utilisé pour le bruit (semble ne pas fonctionner) ;
- **beta1, beta2** : les coefficients β_1 et β_2, cf description du modèle (défaut : $\beta_1 = 1$, $\beta_2 = 0$;
- **bw_hrf** : largeur de la gaussienne utilisée pour lisser la HRF (défaut=0, pas de lissage)
- **seuil** : seuil utilisé pour ne pas considérer un voxel comme « fond » ; ce paramètre est à adapter en fonction des données si le fond n'est pas uniformément à 0 (défaut : 0).
- **fracinit** : lors de la classification initiale, un voxel est classé comme activé (resp. désactivé) si le coefficient de corrélation est supérieur à fracinit×maxcorrel, (resp. inférieur à fracinit×mincorrel), où maxcorrel et mincorrel sont respectivement les plus grands et plus petits coefficients de corrélation calculés (défaut=0.5) ;
- **seuilinit** : si seuilinit≠-1, fracinit est ignoré et la règle de classification initiale suivante est utilisée : si le coefficient de corrélation est supérieur à seuilinit (resp. inférieur à -seuilinit) le voxel est classé comme activé (resp. désactivé) (défaut=-1) ;
- **bloc** : si présent, les onsets sont spécifiés en mode bloc (défaut : mode évènements).

A.2.2 Conditions

Une ligne par condition, avec une chaîne de caractères nommant la condition, puis les instants d'arrivée des onsets (en secondes), terminée par un point.

A.2.3 Exemples de fichier de design

```
fichier /tmp/aaa.nii
tr 1
duree 360
beta1 1
hmax 60
ev1 0 120 240 .
ev2 60 180 300 .
```

Exemple utilisé pour les données de la section 3.5.3 :

```
fichier /data/map284/1801/smoothed/srua1801_
bloc
seuil 1000
beta1 1
beta2 0
tr 2.5
duree 288
image1 6
hrfsample 1
bw_hrf 20
fracinit 0.9
c1 0 12 36 48 72 84 108 120 144 166 190 202 226 238 262 274 .
c2 12 24 48 60 84 96 120 132 166 178 202 214 238 250 274 286 .
c3 24 36 60 72 56 108 132 144 178 190 214 226 250 262 .
```

B Deuxième implémentation

Nous avons implémenté, toujours en C++ – mais sans dépendance sur la librairie LAPACK++ – les deuxième et troisième approches décrites dans ce mémoire. Nous avons également implenté pour le second modèle la méthode d'inférence ICM, afin de pouvoir comparer les résultats obtenus suivant les deux méthodes d'inférence. L'implémentation de la troisième approche (modèle auto-gaussien) est encore à l'état de prototype et n'est pas utilisable sur des données réelles (pour des raisons de performance).

En plus de ressortir les résultats sous forme de fichiers Nifti, le programme ressort un « rapport » au format HTML comprenant un graphe de la HRF estimée, ainsi que les valeurs des différents contrastes, superposées avec l'image anatomique si celle-ci est disponible, ou par défaut, avec la moyenne des images fonctionnelles.

Les répertoires de sortie sont définis par des macros `#define` du fichier `lib.hpp` :

- `CACHE_DIR` : répertoire contenant les images en cache ;
- `FILTERED_DIR` : répertoire contenant les images après application du filtre passe-haut ;
- `OUT_DIR` : répertoire contenant les cartes résultats au format Nifti ;
- `REPORT_DIR` : répertoire contenant les rapports au format HTML.

Les paramètres sont toujours lus dans un fichier de design ; la description de l'expérience suit le même format que dans la première implémentation. Voici la liste des paramètres ainsi que leur signification :

- **fichier** : le nom du fichier à partir duquel lire les données 4D ; si ce paramètre se termine par un underscore (`_`) une série d'images 3D est lue ;
- **freq** : fréquence de coupure du filtre passe-haut appliqué sur les données (mettre au moins à 1 pour retirer la composante continue) ; le paramètre à donner est le numéro de la dernière composante DCT à couper.
- **image1** : si une série d'images 3D est lue, numéro de la première image à traiter (défaut 1) ;
- **tr** : temps de répétition en secondes ;
- **duree** : durée de l'expérience en secondes ;
- **hrfsample** : période d'échantillonnage de la HRF en secondes (défaut=tr) ;
- **hrfcano** : si présent, la HRF n'est pas estimée (défaut : la HRF est estimée) ;
- **hmax** : taille de la HRF à estimer en secondes (défaut : 28) ;
- **ar1** : si présent, un modèle AR(1) est utilisé pour le bruit (semble ne pas fonctionner) ;
- **beta1** : pour le modèle auto-gaussien uniquement, coefficient β_3 ; pour le modèle mélange de gaussiennes, beta1 est estimé et ce paramètre est ignoré ;
- **beta2** : ignoré ;
- **bw_hrf** : largeur de la gaussienne utilisée pour lisser la HRF (défaut=0, pas de lissage)
- **seuil** : seuil utilisé pour ne pas considérer un voxel comme « fond » ; ce paramètre est à adapter en fonction des données si le fond n'est pas uniformément à 0 (défaut : 0).

- **fracinit** : ignoré ;
- **seuilinit** : ignoré ;
- **scoreinit** : t-score minimal pour la classification initiale comme voxel activé (défaut : 1) ;
- **scoreinitn** : t-score maximal pour la classification initiale comme voxel désactivé (défaut : -scoreinit) ;
- **bloc** : si présent, les onsets sont spécifiés en mode bloc (défaut : mode évènements) ;
- **fonctio** : chemin d'accès vers l'image anatomique pour superposition des résultats (défaut : superposer la moyenne des images fonctionnelles) ;
- **icm** : utiliser ICM (défaut : ne pas utiliser) ;
- **ppseuil** : seuil d'affichage des contrastes dans les pages de résultats ;
- **masque** : chemin d'accès vers le masque de traitement, au format Nifti ;
- **dumpfitted** : ressortir le modèle ajusté ;
- **dumpaz** : ressortir les fichiers Nifti contenant les paramètres estimés à chaque itération de l'algorithme ;
- **lambda** : rapport minimal entre les deux conditions à comparer ;
- **ag** : utiliser le modèle auto-gaussien (expérimental).