A C A D É M I E
D E
G R E N O B L E

Institut National Polytechnique de Grenoble

— cohabilité U.J.F., U.P.M.F., STENDHAL —

Rapport de Master II Rercherche

SPÉCIALITÉ SCIENCES COGNITIVES

MÉTHODES STATISTIQUES MULTIVARIÉES EN IRMF

par

Pierre LAFAYE DE MICHEAUX

sous la direction de

Michel DOJAT et Christian JUTTEN

Soutenu le **25 juin 2007** devant le jury composé de

M. DOJAT	Ingénieur de Recherche, INSERM, UJF, Grenoble
A. GUÉRIN-DUGUÉ	Professeur, INPG, Grenoble
C. JUTTEN	Professeur, INPG, Grenoble
M. DOHEN	Maître de Conférences, ENSERG, Grenoble

Stage réalisé au sein du **Grenoble Institut des Neurosciences** (GIN),
Équipe de Neuroimagerie Fonctionnelle et Métabolique,
U836 INSERM-UJF-CEA-CHU

Résumé

L'imagerie par résonance magnétique fonctionnelle (IRMF) est une technique de neuroimagerie récente permettant de localiser l'activité neuronale avec une grande précision spatiale ainsi qu'une bonne précision temporelle. Pour détecter des zones actives dans le cerveau, cette méthode se base sur les changements locaux d'oxygénation du sang qui sont reflétés par de petites variations dans un certain type d'images obtenues par résonance magnétique. La possibilité de dresser une carte fonctionnelle cérébrale de façon non invasive fournit de nouvelles opportunités pour déméler les mystères du cerveau humain.

Dans ce mémoire, nous décrivons quelques méthodes non paramétriques d'analyse multivariée des données IRMF : l'Analyse en Composantes Principales, l'Analyse en Composantes Indépendantes et la Poursuite de Projection. Nous tâchons aussi d'expliquer le lien existant entre ces méthodes, les différentes visions que l'on peut en avoir et abordons les aspects spatiaux et temporels sous-jacents.

Nous avons également élaboré un programme informatique permettant de simuler, de façon extrêmement simplifiée, des signaux IRMF cérébraux. Cet outil nous permet de générer artificiellement des données de type IRMF pour lesquelles nous contrôlons un grand nombre de paramètres. Il servira à comparer quantitativement les méthodes statistiques présentées.

Nous appliquons aussi ces différentes méthodes statistiques sur un jeu de données réelles issues d'une expérience IRMF en vision humaine.

Enfin, nous proposons différentes pistes de recherche qui pourront être explorées pour donner une suite à ce travail préliminaire.

Table des matières

Table des figures

Notations

Scalaire, vecteur, matrice. Ni réalisation, ni aléatoire. Italique cursive.
x, $\boldsymbol{x}$, $\mathcal{X}$

Realisations : scalaire, vecteur, matrice. Italique romaine.
x, $\boldsymbol{x}$, X

Aléatoire : scalaire, vecteur, matrice. Droit romaine.
x, $\mathbf{x}$, X

Autres ensembles ou fonctions :
$\mathbf{I}, \mathbb{N}$

$\mathrm{cov}, \mathrm{var}$: covariance, variance échantillonnale

$\mathrm{Cov}, \mathrm{Var}$: covariance, variance d'une variable ou d'un vecteur aléatoire

det : déterminant d'une matrice.

$\otimes$: produit tensoriel

$\dot{X}$: matrice des données centrée en lignes et en colonnes

Remerciements

Un certain nombre de personnes qui ont facilité l'écriture de ce rapport méritent d'être citées ici.

Tout d'abord, je tiens à remercier chaleureusement Keith Worsley, Professeur à l'université McGill de Montréal, qui m'a ouvert les yeux sur le passionnant sujet de l'imagerie cérébrale et de toutes les questions statistiques que l'on peut s'y poser. Le post-doctorat que j'ai été amené à interrompre pour occuper mon poste à l'Université Pierre Mendès France aurait sûrement été très enrichissant à ses côtés. C'est aussi à l'occasion de sa venue à Grenoble, où il a accepté mon invitation à venir donner un séminaire, que j'ai eu la chance d'entamer des discussions, germes de collaborations futures, avec Monica Bacciu et Michel Dojat. Monica m'a permis de passer, en tant que sujet, ma première IMRF et j'ai pu ainsi observer pour la première fois sur l'écran de mon PC l'intérieur de ma tête. C'est assez magique pour un mathématicien habitué à jouer avec des équations abstraites de se retrouver brutalement projeté dans les méandres de son propre cortex !

C'est d'ailleurs dans l'un de ces mystérieux giri qu'a dû naître l'idée de faire un Master en Neuroscience. Après un bon délai de réflexion, le Master en Sciences Cognitives de Grenoble m'a paru être le plus en adéquation avec mes aspirations, mes compétences et mes contraintes familiales et professionnelles du moment. Et je ne regrette pas mon choix : quels cours passionnants ! Cette année d'études, partagée avec cinq autres étudiants très sympatiques et motivés (Maude Fantoni, Benjamin Lacroix, Clément Moulinfrier, Lucile Rapin et Emmanuelle Reuter) a été très agréable même si j'avais parfois l'impression d'être engagé dans un marathon pour parvenir à concilier toutes mes activités. Je dois dire que la secrétaire du Master, Razika Hammache, était très attentionnée et nous a bien facilité la tâche.

J'ai ensuite été accueilli dans l'Équipe de Neuroimagerie Fonctionnelle et Métabolique par Michel Dojat chez qui j'ai fait un stage passionnant. Il a passé de nombreuses heures avec moi pour partager ses grandes connaissances du domaine et j'ai beaucoup appris à son contact : merci Michel ! Merci aussi à toi Christian pour toutes nos discussions sur les subtilités de l'ACI et ton accueil toujours amical !

Ces trois mois de stage passés dans les sous-sols de l'unité m'auront aussi permis de rencontrer deux étudiants (Nicolas Pannetier et Frédéric Grouiller) avec qui nous auront beaucoup échangé. Je leur souhaite beaucoup de réussite pour la poursuite de leurs études et la suite de leur carrière.

Je tiens enfin à remercier l'Université Pierre Mendès France de m'avoir accordé un Congé pour Recherche et Conversion Thématique d'un semestre, mes collègues du Département de Statistique et Traitement Informatique de Données pour avoir supporté la surcharge de travail due à mon absence et la *SAGAG team* pour cette mise entre parenthèse de nos projets de recherche.

À Domi, Luka et Mathias ...

1 Introduction

L'imagerie fonctionnelle cérébrale par résonance magnétique (IRMF) est une technique non-invasive qui permet l'étude des processus cognitifs chez l'homme sain ou malade. Une expérience en IRMF consiste généralement à soumettre le sujet à des stimuli, définis relativement à la fonction cognitive à explorer, et formés de phases alternant condition de repos et condition d'activation. Un ensemble d'images volumiques est acquis pendant ces périodes. L'exploitation de ces images consiste à déterminer quelles sont les aires du cortex cérébral activées par les stimuli. La détection des voxels qui ont un comportement déterminé par le stimulus n'est pas une opération facile car les signaux obtenus sont faibles (l'activation représente 1 à 2 pourcents du signal basal). Il est alors nécessaire d'utiliser des techniques statistiques. Deux classes de techniques sont disponibles, celles où un modèle est défini *a priori* et des outils statistiques sont utilisés pour déterminer la validité du modèle face aux données (voir [KF94]) et celles à base de séparation de sources, qui permettent de retrouver quels signaux sont contenus dans le mélange observé [MMB$^+$98, MHS03]. La première, largement utilisée notamment au moyen du logiciel SPM [Fri03b], permet de faire des tests d'hypothèses dans un cadre théorique bien établi. La seconde, potentiellement intéressante, limite l'injection d'*a priori* sur le type de signaux à rechercher et le type de bruit mais se heurte à diverses difficultés pratiques.

Dans ce travail, nous évaluons le potentiel des méthodes statistiques de séparation de sources que sont l'Analyse en Composantes Indépendantes (ACI) et la Poursuite de Projection (PP). Ces deux méthodes utilisent la notion d'indépendance statistique, intimement reliée à la non gaussianité d'une distribution. Elles peuvent être vues, d'une certaine manière, comme un raffinement des méthodes plus anciennes d'Analyse en Composantes Principales (ACP). Dans une première partie, nous présentons brièvement les données récoltées dans le cadre d'une expérience d'IRMF. Dans les sections 3 et 4, nous introduisons les notions d'indépendance, de corrélation et d'entropie. Suivent ensuite une présentation assez exhaustive de la méthode d'Analyse en Composantes Principales, dans sa version spatiale et dans sa version temporelle, puis une présentation du problème de sépration de sources et de sa résolution par l'Analyse en Composantes Indépendantes ainsi que par la méthode de Poursuite de Projection. Cette analyse en profondeur de l'ACI dans le cadre de l'exploration des signaux IRMF nous a conduit a proposer un résultat mathématique intéressant (Résultat 1 en page 44 des annexes) et qui, à notre connaissance, n'a jamais été utilisé en ACI. Il permet en effet d'effectuer de façon astucieuse une ACI temporelle, réputée comme impraticable pour l'analyse des données d'IRMF compte tenu de la grande dimension des matrices impliquées dans ce contexte. Nous avons aussi dressé un bilan des axes de recherche à approfondir qui permettront de bâtir des outils statistiques robustes afin que l'ACI soit plus largement utilisée en neurosciences.

Les méthodes que nous avons décrites consistent à retrouver dans les données des cartes source spatialement indépendantes ainsi que leur décours temporel associé. Elles ont été mises en oeuvre à la fois sur des données simulées et sur un jeu de données réelles collectées lors d'une expérience d'IRMF en vision humaine. Le principal intérêt du mini simulateur que nous proposons réside dans le fait qu'il permet d'appliquer rapidement différentes méthodes statistiques en vue de la comparaison de leur efficacité et de leurs éventuels défauts. Nous pouvons générer plusieurs signaux spatio-temporels mélangés dans un modèle résolument simplifié d'une coupe cérébrale et évaluer comment diverses méthodes statistiques parviennent à déméler les signaux temporels originaux ainsi que leur localisation spatiale. La description de ce simulateur se trouve à la section D des annexes (page 51) et sa lecture nous paraît utile pour permettre au lecteur de bien comprendre les objectifs et les résultats des méthodes statistiques que nous examinons. La section E des annexes illustre les résultats que nous avons obtenus sur un jeu de données réelles par application de la méthode d'Analyse en Composantes Indépendantes ainsi que par l'application plus classique du modèle de régression linéaire via le logiciel SPM [Fri03b]. Tous ces résultats ont été reportés dans les annexes par soucis de concision. Le lecteur pourra également y trouver une brève introduction à l'imagerie cérébrale (section A), quelques résultats mathématiques intéressants (section B) et des algorithmes (section C) permettant d'implémenter les méthodes statistiques décrites dans le corps du rapport.

Nous avons par ailleurs codé plusieurs fonctions développées grâce au logiciel R [R D06], abondamment utilisé dans le milieu statistique universitaire. Ces fonctions, disponibles à la section F, devraient permettre, nous l'espérons, de faciliter les échanges entre la communauté des chercheurs en statistique et la communauté des chercheurs en neuroimagerie. Ces fonctions permettent entre autres de lire et écrire facilement des jeux de données au format d'image Analyze, de visualiser des images anatomiques et fonctionnelles, et bien sûr d'effectuer les analyses statistiques décrites dans ce rapport.

2 Les données et leur traitement statistique

Les données acquises suite à une expérience d'IRMF (mesures de l'effet BOLD) consistent en des images tridimensionnelles (parallélépipède rectangle) du cerveau ou d'une région cérébrale prises à t_m valeurs de temps et discrétisées en un certain nombre v_m de voxels. Cette information est stockée dans t_m tableaux à trois dimensions, chacun possédant v_m cases que l'on appelle des voxels (équivalent 3D du pixel).

Les valeurs stockées dans ces tableaux sont des mesures de radiométrie codées en niveaux de gris et variant entre 0 et $2^{16} - 1$.

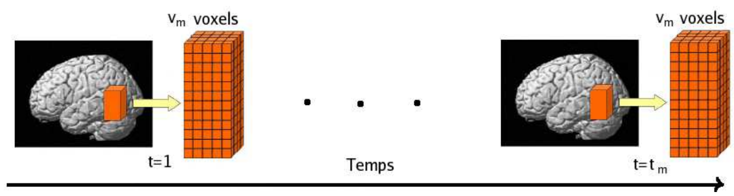


Fig. 1 – Données récoltées lors d'une expérience d'IRMF

Remarque 1. *Une des difficultés dans l'analyse statistique des données de ce type provient du nombre très important de données à gérer simultanément. Deux approches sont possibles pour réduire la taille de ces données. Il est par exemple possible d'effectuer un dépliage du cortex [War02] et donc de considérer que les données peuvent être stockées dans une seule matrice (de taille $t_m \times$ le nombre de voxels du dépliage). C'est une technique intéressante qui permet de réduire le nombre de données. Elle est toutefois surtout utile pour l'étude des aires présentant une topie pour laquelle une délinéation précise est possible (par exemple les aires rétinotopiques où l'on sait faire de la délinéation des aires visuelles).*
Une autre option intéressante est d'effectuer un masque du cerveau afin de ne pas conserver les parties de l'image situées en dehors du cerveau.

Les méthodes de traitement statistique de données que nous allons présenter (Analyse en Composantes principales (ACP), Analyse en Composantes Indépendantes (ACI), Poursuite de Projection (PP)) nécessitent de travailler sur des matrices à deux dimensions. Par conséquent, quitte à négliger, en quelque sorte, la répartition spatiale des voxels (ne pas la négliger impliquerait d'utiliser des méthodes de calcul beaucoup plus coûteuses comme par exemple la théorie des champs Markoviens), on peut écrire les données sous la forme $x_v(t) = x_v^t$ $(1 \leq v \leq v_m,\ 1 \leq t \leq t_m)$. Deux points de vue sont alors possibles. On peut considérer soit que l'on dispose de t_m cartes spatiales d'activations du cerveau (*spatial maps*) de longueur v_m, soit de v_m signaux temporels de longueur t_m (*time courses*), que l'on pourra regrouper dans une matrice de dimension $t_m \times v_m$ ou $v_m \times t_m$, respectivement. Les méthodes s'intéressant au premier point de vue seront qualifiées de spatiales, celles s'intéressant au second point de vue de temporelles.

Toutes les méthodes statistiques utilisées dans le cadre de cet ouvrage seront appliquées sur la matrice des données (ou sa transposée) préalablement centrée à la fois en colonnes puis en lignes, en prenant garde que la matrice résultante possède toujours des variables de moyenne nulle. La raison pour centrer les données suivant le domaine temporel, c'est-à-dire d'enlever l'image moyenne, est que celle-ci n'est pas du tout intéressante. Ce qui nous intéresse ce sont les variations autour de la moyenne. Si l'on ajoutait un décalage fixé à chaque série temporelle, on ne voudrait pas générer une décomposition différente ; on veut que la méthode employée soit insensible à la vraie valeur moyenne. C'est pourquoi la moyenne pour chaque décours temporel (la moyenne image globale) est retranchée. De façon similaire, le décours temporel moyen n'est pas très informatif (il est obtenu en moyennant sur l'espace, en pondérant chaque voxel de façon similaire) et il sera retranché. Un autre avantage du centrage est de nature algorithmique car de nombreux algorithmes nécessitent cette condition qui simplifie grandement les calculs. Voilà l'équation reliant la matrice originale et la matrice doublement centrée. Les vecteurs moyenne ligne (centrage des colonnes) et colonne (centrage des lignes) sont calculés successivement.

$$\dot{X} = \left[X - \mathbf{1}_{t_m}\boldsymbol{v}_{mean}^{\mathsf{T}}\right] - \boldsymbol{t}_{mean}\mathbf{1}_{v_m}^{\mathsf{T}}. \tag{2.1}$$

$$\dot{X}^{\mathsf{T}} = \left[X - \boldsymbol{t}_{mean}\mathbf{1}_{v_m}^{\mathsf{T}}\right] - \mathbf{1}_{t_m}\boldsymbol{v}_{mean}^{\mathsf{T}}. \tag{2.2}$$

Par ailleurs, toutes ces méthodes font appel aux notions d'indépendance et de corrélation que nous introduisons dans la section suivante. De ce point de vue, les variables gaussiennes présentent une certaine particularité que nous décrivons. Nous définissons également la notion d'entropie qui nous sera elle aussi utile par la suite.

3 Indépendance et corrélation

Il est primordial de bien comprendre les notions d'indépendance et de corrélation qui constituent le coeur des méthodes statistiques utilisées dans ce mémoire. Une façon simple de comprendre l'indépendance repose sur la notion plus familière de corrélation. La corrélation entre deux variables aléatoires $\textsc{x}$ et $\textsc{y}$ est

$$\rho(\textsc{x},\textsc{y}) = \frac{\mathrm{Cov}(\textsc{x},\textsc{y})}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}}, \tag{3.1}$$

où σ_{X} et σ_{Y} sont les écarts-types de $\textsc{x}$ et $\textsc{y}$ respectivement, et où

$$\mathrm{Cov}(\textsc{x},\textsc{y}) = \mathbf{E}[\textsc{xy}] - \mathbf{E}[\textsc{x}]\mathbf{E}[\textsc{y}] \tag{3.2}$$

est la covariance entre $\textsc{x}$ et $\textsc{y}$. La quantité $\mathbf{E}[\textsc{xy}]$ est un moment d'ordre 2 de la distribution jointe $F_{(\mathrm{X},\mathrm{Y})}$. Les quantités $\mathbf{E}[\textsc{x}]$ et $\mathbf{E}[\textsc{y}]$ sont les premiers moments respectivement des distributions marginales F_{X} et F_{Y}.

La corrélation est donc simplement une forme de covariance qui a été normalisée pour varier entre -1 et +1. Notons que si deux variables $\textsc{x}$ et $\textsc{y}$ sont non corrélées alors $\rho(\textsc{x},\textsc{y}) = \mathrm{Cov}(\textsc{x},\textsc{y}) = 0$, bien que $\rho(\textsc{x},\textsc{y})$ et $\mathrm{Cov}(\textsc{x},\textsc{y})$ ne soient pas égales en général.

La covariance ne capture pas tous les types de dépendance entre $\textsc{x}$ et $\textsc{y}$, alors que des mesures statistiques visant à détecter l'indépendance le font.

Comme la covariance, l'indépendance peut être définie en terme des moments de la distribution jointe $F_{(\mathrm{X},\mathrm{Y})}$. On a établi que $\textsc{x}$ et $\textsc{y}$ sont non corrélées si et seulement si

$$\mathbf{E}[\textsc{xy}] - \mathbf{E}[\textsc{x}]\mathbf{E}[\textsc{y}] = 0. \tag{3.3}$$

En utilisant une forme généralisée de la covariance impliquant les puissances de $\textsc{x}$ et $\textsc{y}$, on peut montrer de façon théorique que $\textsc{x}$ et $\textsc{y}$ sont statistiquement indépendantes si et seulement si

$$\mathbf{E}[\textsc{x}^p\textsc{y}^q] - \mathbf{E}[\textsc{x}^p]\mathbf{E}[\textsc{y}^q] = 0 \tag{3.4}$$

pour tous les entiers positifs p et q. Ainsi, alors que la covariance utilise $p = q = 1$, toutes les valeurs entières positives de p et q sont implicites dans les mesures de dépendance.

La similarité formelle entre les mesures de dépendance et de covariance peut être interprétée de la façon suivante. Alors que la covariance mesure le degré de covariation linéaire entre X et Y, les mesures de dépendance s'intéressent à la covariation linéaire entre X^p et Y^q, $\forall p, q \in \mathbb{N}^*$. Ainsi, les mesures de dépendance peuvent être considérées implicitement comme une forme généralisée de la covariance, qui mesurent la covariation linéaire entre des fonctions non linéaires de deux variables.

Il est bien connu que l'indépendance entraîne la non corrélation. Montrons sur un exemple théorique simple qu'à l'inverse la non corrélation entre X et Y n'implique pas leur indépendance.

Exemple 1. *Étant donnée une variable aléatoire continue* Z *de loi* $Unif[0, 2\pi]$*, on peut définir les variables* $\mathrm{X} = \sin(\mathrm{Z})$ *et* $\mathrm{Y} = \cos(\mathrm{Z})$*. Il est possible de montrer que* $\mathrm{Cov}(\mathrm{X}, \mathrm{Y}) = 0$*. Cependant, la mesure de la covariation linéaire entre les variables* X^p *et* Y^q *pour* $p = q = 2$ *est*

$$\mathbf{E}[\mathrm{X}^2\mathrm{Y}^2] - \mathbf{E}[\mathrm{X}^2]\mathbf{E}[\mathrm{Y}^2] \approx -0.125. \tag{3.5}$$

Cela correspond à une corrélation entre X^2 *et* Y^2 *de* -1*. Ainsi, alors que la corrélation entre* $\mathrm{X} = \sin(\mathrm{Z})$ *et* $\mathrm{Y} = \cos(\mathrm{Z})$ *est zéro, le fait que la valeur de* X *puisse être prédite à partir de* Y *est implicite dans les valeurs non nulles des moments d'ordre supérieur de la distribution de* (X, Y)*.*

Illustrons ceci sur un graphique de variables simulées.

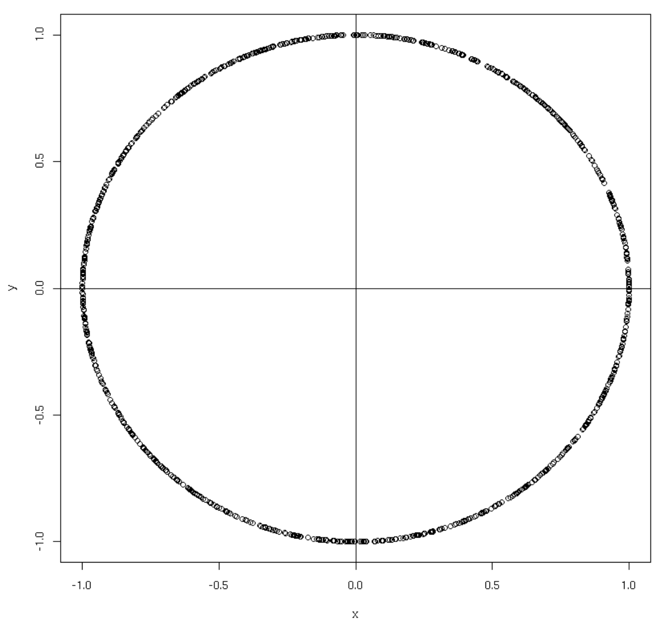


Fig. 2 – On définit la variable aléatoire Z de loi $Unif[0, 2\pi]$, et les variables $\mathrm{X} = \sin(\mathrm{Z})$ et $\mathrm{Y} = \cos(\mathrm{Z})$. Les variables X et Y sont non corrélées mais dépendantes. Un échantillon $(\boldsymbol{X}, \boldsymbol{Y})$ de taille 1000 est représenté.

La covariance échantillonnale de $\boldsymbol{x}$ *et* $\boldsymbol{y}$ *est donnée par*

$$\operatorname{cov}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{\boldsymbol{x}})(y_i - \bar{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \bar{\boldsymbol{x}}\,\bar{\boldsymbol{y}} \tag{3.6}$$

où $\bar{\boldsymbol{x}}$ *et* $\bar{\boldsymbol{y}}$ *sont les moyennes des réalisations contenues dans les vecteurs* $\boldsymbol{x}$ *et* $\boldsymbol{y}$ *respectivement. Sans perte de généralité, on peut supposer que les variables sont centrées (ce qui est le cas ici) et l'on a donc*

$$\operatorname{cov}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} x_i y_i. \tag{3.7}$$

On constate sur la figure 2 qu'il y a une symétrie de la distribution des données par rapport à l'axe des abscisses et des ordonnées du repère dont l'origine est le centre de gravité du nuage de points. Par conséquent, quasiment chacun des termes $x_i y_i$ *de la somme précédente est annulé par un autre terme de même valeur absolue mais de signe contraire et l'on a* $\operatorname{cov}(\boldsymbol{x},\boldsymbol{y}) \approx 0$*.*
La corrélation étant nulle si et seulement si la covariance est nulle, on constate donc sur ce graphique que les variables X *et* Y *sont non corrélées.*

Par contre, il y a une dépendance entre les variables aléatoires X *et* Y *puisque la connaissance d'une réalisation* x *de* X *donne des indications sur les valeurs possibles de* Y*. Ainsi, si* x *est voisin de 1,* y *sera proche de* 0 *alors que si* x *est voisin de 0,* y *sera proche de* 1 *ou de* -1*.*
Dans cet exemple particulier, la dépendance est d'ailleurs très forte puisque le fait que $x^2 + y^2 = 1$ *nous dit que la connaissance de l'une des variables suffit à connaître l'autre (au signe près).*

Pour résumer, on retiendra que la non corrélation correspond à une symétrie de la distribution des données par rapport à l'un des axes du repère passant par le centre de gravité du nuage. D'un autre côté, l'indépendance correspond à l'égalité des distributions conditionnelles de $\mathrm{X}|\mathrm{Y} = y$ et, intuitivement, on dira que X et Y sont indépendantes si la connaissance de la valeur de l'une des variables ne peut aider en rien pour prédire la valeur de l'autre variable.

On a montré que la covariance et l'interdépendance entre deux variables aléatoires X et Y sont définies en terme des moments de la densité de leur loi jointe. Cependant, toute variable gaussienne X est spéciale dans la mesure où elle est complètement spécifiée par ses deux premiers moments $\mathbf{E}[\mathrm{X}]$ et $\mathbf{E}[\mathrm{X}^2]$. C'est-à-dire que les valeurs de tous les moments d'ordre supérieur sont implicites dans la valeur du moment d'ordre 2 d'une distribution gaussienne. Ainsi, si la covariance $\mathbf{E}[\mathrm{XY}] - \mathbf{E}[\mathrm{X}]\mathbf{E}[\mathrm{Y}]$ entre deux gaussiennes est zéro alors on peut montrer que la quantité $\mathbf{E}[\mathrm{X}^p\mathrm{Y}^q] - \mathbf{E}[\mathrm{X}^p]\mathbf{E}[\mathrm{Y}^q]$ est aussi nulle pour toutes les valeurs entières supérieures ou égales à 2 de p et q. Par l'équation (3.4), on sait que de telles variables sont statistiquement indépendantes, et il s'ensuit donc que des variables gaussiennes non corrélées sont aussi indépendantes.
Cependant, des variables non gaussiennes qui sont non corrélées ne sont pas, en général, indépendantes (cf. exemple 1). Ainsi, pour des variables non gaussiennes, la dépendance entre X et Y apparaît seulement dans les moments d'ordre supérieur.

4 Entropie

Intuitivement, l'entropie de Shannon peut être vue comme un indicateur numérique mesurant la quantité d'incertitude liée à un événement aléatoire, ou plus précisément à sa distribution. Une autre manière de la voir est comme la quantité d'information portée par un signal : l'information fournie par chaque nouvel événement (après sa réalisation) est égal à l'incertitude sur cet événement (avant sa réalisation).

Par exemple, imaginons une urne contenant plusieurs boules de différentes couleurs, dont on tire une boule au hasard (avec remise). Si toutes les boules ont des couleurs différentes, alors notre incertitude sur le résultat d'un tirage est maximale. En particulier, si nous devions parier sur le résultat d'un tirage, nous ne pourrions pas privilégier un choix plutôt qu'un autre. Par contre, si une certaine couleur est plus représentée que les autres (par exemple si l'urne contient davantage de boules rouges), alors notre incertitude est légèrement réduite : la boule tirée a plus de chances d'être rouge. Si nous devions absolument parier sur le résultat d'un tirage, nous miserions sur une boule rouge. Ainsi, révéler le résultat d'un tirage fournit davantage d'information dans le premier cas que dans le second, parce que l'entropie du "signal" (calculable à partir de la distribution statistique) est plus élevée.

Prenons un autre exemple : considérons un texte en français codé comme une chaîne de lettres, d'espaces et de ponctuations (notre signal est donc une chaîne de caractères). Comme la fréquence de certains caractères n'est pas très élevée (ex : 'z'), tandis que d'autres sont très communs (ex : 'e'), la chaîne de caractères n'est pas si aléatoire que ça. D'un autre côté, tant qu'on ne peut pas prédire quel est le caractère suivant, d'une certaine manière, cette chaîne est aléatoire. L'entropie est une mesure de cet aléa suggérée par Shannon.

On peut considérer qu'un événement de faible probabilité apporte plus d'information qu'un événement de forte probabilité.
En termes simples, moins une observation est probable, plus son observation est porteuse d'information. Par exemple, lorsqu'un journaliste commence le journal télévisé par la phrase "Bonsoir", ce mot, qui présente une forte probabilité, n'apporte que peu d'information. En revanche, si la première phrase est, par exemple "La France a peur", sa faible probabilité fera que l'auditeur apprendra qu'il s'est passé quelque chose, et, partant, sera plus à l'écoute.

On cherche maintenant à formaliser cette notion.

4.1 Cas discret

Soit $(\Omega, \mathbf{A}, \mathbf{P})$ un ensemble probabilisé (Ω est l'ensemble des événements élémentaires, $\mathbf{A}$ est la tribu des événements dont on peut calculer la probabilité $\mathbf{P}$) et soit $\mathrm{x} : \Omega \rightarrow \mathbb{R}$ une variable aléatoire discrète prenant les valeurs $x_1, \ldots, x_n$ avec les probabilités $p_1, \ldots, p_n$. On note $\Omega_i = \mathrm{x}^{-1}(x_i)$ et on a $p_i = Pr[\mathrm{x} = x_i] = \mathbf{P}[\mathrm{x}^{-1}(x_i)] = \mathbf{P}[\Omega_i]$. Les Ω_i forment une partition de Ω.

À tout événement (pas nécessairement élémentaire) est associée l'une des valeurs x_i. Il est donc inclus dans Ω_i.

On peut alors définir la fonction information f de $\mathbf{A}$ dans $[0, \infty]$ qui associe à tout événement $\mathbf{a} \in \mathbf{A}$ la valeur $-\mathrm{Log}(p_i)$, si $\mathbf{a} \in \Omega_i$.

On peut voir que $f(\mathbf{a})$ est grand si $\mathbf{a}$ appartient à un ensemble de petite mesure de probabilité (plus d'information) et on dira que $f(\mathbf{a})$ mesure l'information au point $\mathbf{a}$. On définira l'entropie $H(\mathrm{x})$ de la variable aléatoire x comme la moyenne (espérance mathématique) de la fonction f :

$$H(\mathrm{x}) = \mathbf{E}[f(\mathrm{x})] = -\sum_{i=1}^{n} p_i \mathrm{Log}(p_i) = \int f(\omega) d\mathbf{P}(\omega) \text{ avec } f(\omega) = -\mathrm{Log}(p_i) \text{ si } \omega \in \Omega_i. \tag{4.1}$$

On peut remarquer certaines caractéristiques de cette formule :

- La valeur de H est maximale pour une distribution uniforme, c'est-à-dire quand tous les états ont la même probabilité.
- Toutes choses étant égales par ailleurs, H augmente avec le nombre d'états possibles (ce qui traduit l'intuition que plus il y a de choix possibles, plus l'incertitude est grande).

4.2 Cas continu

La définition de l'entropie pour une variable discrète peut être généralisée à des variables ou des vecteurs aléatoires continus, et dans ce cas on parle souvent d'entropie différentielle. L'entropie différentielle $H(\mathrm{x})$ d'une variable aléatoire x de densité $p_{\mathrm{X}}(.)$ est définie par

$$H(\mathrm{x}) = -\int p_{\mathrm{X}}(x) \mathrm{Log} p_{\mathrm{X}}(x) dx. \tag{4.2}$$

L'entropie différentielle peut être interprétée comme une mesure de l'incertitude de la même façon que l'entropie. Si la variable aléatoire est concentrée sur certains intervalles de petite taille, son entropie différentielle est petite. Notons que l'entropie différentielle peut être négative contrairement à l'entropie. Ainsi quand nous parlons de "petite entropie différentielle", celle-ci peut être négative et avoir une grande valeur absolue. On trouve donc encore que l'entropie (différentielle) est petite quand la variable n'est pas très aléatoire mais contenue dans quelques intervalles limités avec une grande probabilité.

La définition de l'entropie différentielle peut être facilement généralisée au cas multidimensionnel :

$$H(\mathbf{x}) = -\int p_{\mathbf{X}}(\boldsymbol{x}) \mathrm{Log} p_{\mathbf{X}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{4.3}$$

Un résultat fondamental énonce qu'une distribution gaussienne a la plus grande entropie possible parmi toutes les distributions ayant la même matrice de covariance. Cela signifie que l'entropie peut être utilisée comme une mesure de l'écart à la normalité. Cela montre aussi que, à variance donnée, la distribution gaussienne est la "plus aléatoire" ou la moins structurée de toutes les distributions. À partir de ce constat, il semble naturel de proposer l'indicateur **J** suivant, appelé Néguentropie

$$\mathbf{J}(\mathbf{x}) = H(\mathbf{x}_{Gauss}) - H(\mathbf{x}) \tag{4.4}$$

où $\mathbf{x}_{Gauss}$ est un vecteur aléatoire gaussien de même matrice de covariance Σ que $\mathbf{x}$:

$$H(\mathbf{x}_{Gauss}) = \frac{1}{2} \mathrm{Log} |\det \Sigma| + \frac{n}{2} [1 + \mathrm{Log} 2\pi] \tag{4.5}$$

où n est la dimension de $\mathbf{x}$. La Néguentropie possède les caractéristiques intéressantes d'être toujours positive et invariante par des transformations linéaires inversibles. Enfin, elle est nulle uniquement si la variable est gaussienne et plus sa valeur est grande plus la variable en question a une distribution qui s'éloigne de la gaussienne.

Notons enfin que l'on peut aussi définir à partir de l'entropie une quantité **Inf**, appelée Information mutuelle entre les n composantes d'un vecteur aléatoire $\mathbf{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)^{\mathsf{T}}$ de densité $p_{\mathbf{X}}$, par la relation suivante :

$$\mathbf{Inf}(\mathrm{x}_1, \ldots, \mathrm{x}_n) = \mathbf{E} \left[\mathrm{Log} \left(\frac{p_{\mathbf{X}}}{\prod_i p_{\mathrm{X}_i}} \right) \right] = \sum_{i=1}^{n} H(\mathrm{x}_i) - H(\mathbf{x}), \tag{4.6}$$

où p_{X_i} est la densité de x_i.
Il est possible de montrer que l'information mutuelle est toujours positive et qu'elle vaut zéro si et seulement si les variables x_i sont indépendantes.

5 Analyse par la méthode des Composantes Principales (ACP)

L'ACP est une méthode statistique qui permet : la représentation graphique de données quantitatives multidimensionnelles, la réduction de dimension d'un jeu de données récoltées sur un grand nombre de variables corrélées, le calcul d'un certain nombre de facteurs expliquant la variabilité présente dans les données.

Cette méthode n'est efficace que s'il existe une certaine corrélation entre les variables (si les variables sont non corrélées, leur injection en entrée d'un algorithme d'ACP nous les redonne en sortie). C'est pourquoi l'inspection visuelle de la matrice des corrélations est toujours un préalable à l'utilisation de ce type de méthodes. Nous donnons ci-dessous un exemple de visualisation d'une telle matrice.

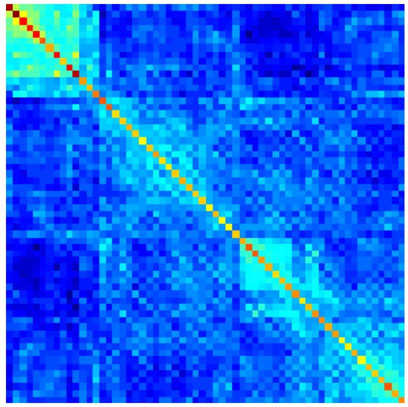


FIG. 3 – Image de la matrice de corrélation des données présentant le degré de corrélation existant entre les variables.

5.1 L'Analyse en Composantes Principales spatiale (ACPs)

On considère que l'on a un vecteur aléatoire $\mathbf{x} = (\mathrm{x}^1, \dots, \mathrm{x}^{t_m})^\intercal$ à t_m composantes et que l'on dispose d'un échantillon de taille v_m de tels vecteurs : $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{v_m}$. On peut donc regrouper les données IRMF dans une matrice X de dimension $t_m \times v_m$:

$$X = [\boldsymbol{x}_1| \dots |\boldsymbol{x}_{v_m}] = \begin{pmatrix} X_1^1 & \dots & X_v^1 & \dots & X_{v_m}^1 \\ \vdots & & \vdots & & \vdots \\ X_1^{t_m} & \dots & X_v^{t_m} & \dots & X_{v_m}^{t_m} \end{pmatrix} = \begin{pmatrix} \text{Carte d'activations } 1 \\ \dots \\ \text{Carte d'activations } T \end{pmatrix} = \begin{pmatrix} \mathrm{v} & | & & | & \mathrm{v} \\ \mathrm{o} & | & & | & \mathrm{o} \\ \mathrm{x} & | & \dots & | & \mathrm{x} \\ \mathrm{e} & | & & | & \mathrm{e} \\ \mathrm{l} & | & & | & \mathrm{l} \\ {}_1 & | & & | & {}_{v_m} \end{pmatrix}. \tag{5.1}$$

La colonne numéro v de X représente les données temporelles dont on dispose pour le voxel v.

L'objectif de l'ACP spatiale (notée ACPs dans la suite) est de rechercher des motifs spatiaux intéressants dans les données IRMF.

Remarque 2. L'écriture de la matrice des données avec les variables en lignes et les individus en colonnes est contraire aux notations standards utilisées dans la communauté statistique. C'est toutefois celle que nous avons choisie ici et dans le reste de ce rapport pour être en accord avec la littérature en neuroimagerie.

Il y a plusieurs façons complémentaires de voir l'ACPs que nous présentons ci-dessous. Ces trois approches aboutissent bien entendu aux mêmes résultats.

Vision de l'ACPs par maximisation de la variance

On cherche à obtenir, dans une version minimaliste de l'ACP, le meilleur résumé unidimensionnel des t_m variables aléatoires $\mathrm{x}^1, \dots, \mathrm{x}^{t_m}$. On va se restreindre à un résumé sous la forme d'une combinaison linéaire (moyenne pondérée) et l'information à résumer ici est la variabilité présente dans les données, mesurée par

un indicateur appelé la variance.
On cherche donc la combinaison linéaire $\mathrm{Y}^1 = \sum_{t=1}^{t_m} w_{1t}\mathrm{X}^t = \boldsymbol{w}_1^\intercal \mathbf{x}$ des t_m variables aléatoires $\mathrm{X}^1, \ldots, \mathrm{X}^{t_m}$ dont la variance est maximale. Les poids de cette combinaison linéaire sont donnés par l'équation

$$\boldsymbol{w}_1 = (w_{11}, \ldots, w_{1t_m})^\intercal = \underset{\{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{t_m})^\intercal;\|\boldsymbol{\alpha}\|=1\}}{\mathrm{Argmax}} \mathrm{Var}\left(\sum_{t=1}^{t_m} \alpha_t \mathrm{X}^t\right). \tag{5.2}$$

La condition $\|\boldsymbol{\alpha}\| = 1$ est nécessaire car sans elle la variance dans la formule ci-dessus pourrait croître sans limite.
La variable aléatoire Y^1 est la moyenne pondérée des X^t qui capture le plus de variabilité dans les données. On appelle cette variable la première composante principale. Le vecteur $\boldsymbol{w}_1$ est appelé le premier axe (ou facteur) principal.
Les composantes principales subséquentes $\mathrm{Y}^2 = \boldsymbol{w}_2^\intercal \mathbf{x}, \ldots, \mathrm{Y}^{t_m} = \boldsymbol{w}_{t_m}^\intercal \mathbf{x}$ qui permettent la construction de résumés multidimensionnels sont obtenues de la même manière mais en rajoutant la contrainte que la j-ème composante doit être non corrélée aux précédentes :

$$\boldsymbol{w}_j = \underset{\{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{t_m})^\intercal;\|\boldsymbol{\alpha}\|=1\}}{\mathrm{Argmax}} \mathrm{Var}\left(\sum_{t=1}^{t_m} \alpha_t \mathrm{X}^t\right) \text{ avec } \boldsymbol{w}_j \text{ non corrélée à } \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{j-1}; \quad \forall j = 2, \ldots, t_m. \tag{5.3}$$

La non corrélation entre deux variables se traduit par leur orthogonalité (dans l'espace des variables). Ainsi, ces autres composantes vont rechercher l'information portée par les données dans d'autres directions orthogonales. Par exemple, la deuxième composante principale Y^2 est aussi une combinaison linéaire de $\mathrm{X}^1, \ldots, \mathrm{X}^{t_m}$ de variance maximale (inférieure à la variance de Y^1) mais qui doit aussi être non corrélée avec la première composante principale Y^1.
La figure 4 illustre ces propos dans le cas de deux variables en montrant comment l'ACP recherche les directions orthogonales "d'allongement maximal" dans les données.

(a)
(b)

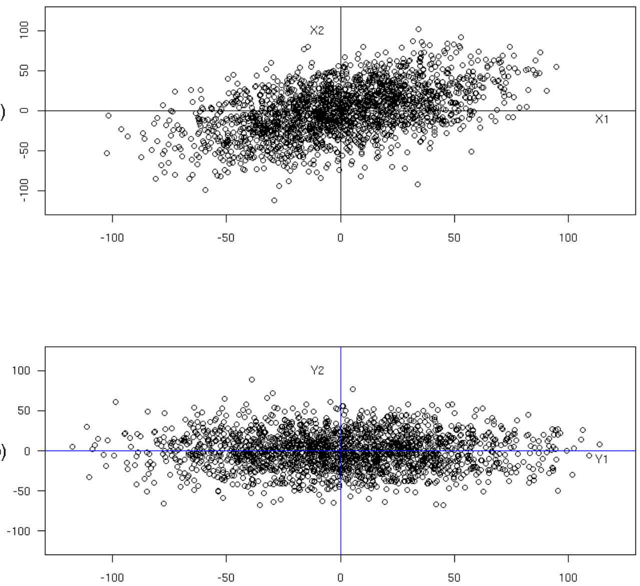


FIG. 4 – (a) Échantillon de taille 2000 provenant d'un vecteur aléatoire $(\mathrm{X}^1, \mathrm{X}^2)^\mathsf{T}$ de densité binormale. (b) Les axes en bleu représentent les directions orthogonales d'allongement maximal de la distribution obtenues par l'ACP.

Notons que la quantité à maximiser dans (5.3) peut s'écrire sous la forme

$$\mathrm{Var}\left(\sum_{t=1}^{t_m} \alpha_t \mathrm{X}^t\right) = \mathrm{Var}\left(\boldsymbol{\alpha}^\mathsf{T}\mathbf{x}\right) = \boldsymbol{\alpha}^\mathsf{T}\mathrm{Var}(\mathbf{x})\boldsymbol{\alpha}. \tag{5.4}$$

Dans la pratique, la matrice inconnue $\mathrm{Var}(\mathbf{x})$ de taille $t_m \times t_m$ sera estimée par la matrice de variance-covariance échantillonnale $S_\mathrm{X} = \frac{1}{v_m}\dot{X}\dot{X}^\mathsf{T}$ et l'obtention des poids dans (5.3) se fera en fait au moyen de l'équation suivante :

$$\hat{\boldsymbol{w}}_1 = \underset{\{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{t_m})^\mathsf{T};\|\boldsymbol{\alpha}\|=1\}}{\mathrm{Argmax}} \boldsymbol{\alpha}^\mathsf{T} S_\mathrm{X}\boldsymbol{\alpha}. \tag{5.5}$$

Un peu de calcul algébrique (voir le Résultat 2 des annexes en page 45) permet de montrer que $\hat{\boldsymbol{w}}_1$ est égal au vecteur propre $\boldsymbol{e}_1$ associé à la plus grande valeur propre λ_1 de S_X et que $\mathrm{Var}(\mathrm{Y}^1) = \lambda_1$. De la même façon, on montre que les poids $\hat{\boldsymbol{w}}_2, \ldots, \hat{\boldsymbol{w}}_{t_m}$ des autres composantes sont égaux aux vecteurs propres subséquents $\boldsymbol{e}_2, \ldots, \boldsymbol{e}_{t_m}$ associés aux valeurs propres $\lambda_2, \ldots, \lambda_{t_m}$ de S_X classées par ordre décroissant, avec $\mathrm{Var}(\mathrm{Y}^i) = \lambda_i$, $2 \leq i \leq t_m$.

On obtient donc les t_m meilleurs (au sens de l'information apportée par la variance) résumés unidimensionnels $\mathrm{Y}^1, \ldots, \mathrm{Y}^{t_m}$, d'un point $\boldsymbol{x}$ donné, par la formule

$$\boldsymbol{Y} = \begin{pmatrix} Y^1 \\ \vdots \\ Y^{t_m} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_1^\mathsf{T} \\ \vdots \\ \boldsymbol{e}_{t_m}^\mathsf{T} \end{pmatrix} \begin{pmatrix} X^1 \\ \vdots \\ X^{t_m} \end{pmatrix} = \mathcal{E}^\mathsf{T}\boldsymbol{X}, \tag{5.6}$$

où $\mathcal{E}$ est la matrice contenant les vecteurs propres de $v_m S_\mathrm{X}$ organisés en colonnes. L'information (en terme de variance) apportée par les Y_i va en décroissant et en se complétant, Y^1 apportant le plus d'information et Y^{t_m} le moins d'information (chaque nouvelle composante apportant une information que n'apportait pas les précédentes). L'ensemble des premières composantes principales apporte bien sûr un résumé multidimensionnel des données.

Par ailleurs, il est bien connu que $\mathcal{E}$ et $\mathcal{E}^\mathsf{T}$ sont des matrices orthogonales ($\mathcal{E}\mathcal{E}^\mathsf{T} = \mathcal{E}^\mathsf{T}\mathcal{E} = \mathcal{I}$) et sont donc des matrices de rotation (voir le Résultat 3 des annexes en page 45). Ainsi, pour obtenir le vecteur $\boldsymbol{Y}$ à partir du vecteur $\boldsymbol{X}$, il suffit d'effectuer une rotation du repère orthogonal d'origine dans les directions d'allongement maximal de la distribution. Cela apparaît nettement sur le graphique de la figure 4.

La matrice $\mathcal{E}$ des vecteurs propres de S_X est la matrice de la nouvelle base (exprimée dans la base canonique) dans laquelle on va "regarder" le nuage. En particulier, les deux premières colonnes de cette matrice (la première colonne est le vecteur propre associé à la plus grande valeur propre, la deuxième colonne est le vecteur propre associé à la deuxième plus grande valeur propre, etc ...) sont les vecteurs qui déterminent (dans la base canonique) les deux axes du plan principal. Ces vecteurs propres sont appelés les axes (ou facteurs) principaux de l'ACPs.

On notera $\mathcal{C}_\mathcal{E}$ la matrice des composantes principales (coordonnées des points individus dans la nouvelle base des vecteurs propres $\mathcal{E}$) :

$$\mathcal{C}_\mathcal{E} = \mathcal{E}^\mathsf{T} X. \tag{5.7}$$

Les coordonnées de l'individu i dans la nouvelle base sont dans la i-ème colonne de cette matrice.

Vision de l'ACPs par recherche de sources non corrélées

On suppose qu'il y a t_m signaux sources $\mathrm{s}_1, \dots, \mathrm{s}_{t_m}$ inconnus qui sont mélangés, au moyen d'une fonction de mélange (linéaire) $\mathcal{A}$, pour donner t_m observations $\mathrm{x}^1, \dots, \mathrm{x}^{t_m}$:

$$\mathbf{x} = \begin{pmatrix} \mathrm{x}^1 \\ \vdots \\ \mathrm{x}^{t_m} \end{pmatrix} = \mathcal{A} \begin{pmatrix} \mathrm{s}^1 \\ \vdots \\ \mathrm{s}^{t_m} \end{pmatrix} = \mathcal{A}\mathbf{s}. \tag{5.8}$$

La matrice de mélange $\mathcal{A}$ est elle aussi supposée inconnue et les sources sont supposées être non corrélées.

On cherche alors à retrouver les sources $\mathbf{s}$ et la matrice de mélange $\mathcal{A}$ en résolvant le système (5.8).
Pour cela, on cherche à obtenir une estimation $\hat{\mathbf{s}}$ des sources en résolvant le problème (P) suivant :

$$(P) \left[\begin{array}{l} \text{Trouver une matrice } \mathcal{B} = \hat{\mathcal{A}}^{-1} \text{ telle que le vecteur } \mathbf{Y} = \hat{\mathbf{s}} = \hat{\mathcal{A}}^{-1}\mathbf{x} = \mathcal{B}\mathbf{x} \text{ soit tel que } \mathrm{Var}(\mathbf{Y}) = \mathcal{D} \\ \text{où } \mathcal{D} \text{ est une matrice diagonale ayant tous ses éléments diagonaux positifs ou nuls.} \end{array} \right.$$

Notons que la condition sur $\mathcal{D}$ (matrice diagonale) traduit de façon explicite le fait que l'on cherche des sources estimées non corrélées.

Il est important de remarquer que la solution à ce problème n'est pas unique mais n'est, en quelque sorte, résolvable qu'à une rotation (matrice orthogonale) près. En effet, si $\mathcal{B}$ est une solution ayant pour matrice diagonale associée $\mathcal{D}$, alors pour toute matrice orthogonale $\mathcal{O}$ et toute matrice diagonale $\mathcal{F}$ ayant tous ses éléments diagonaux positifs ou nuls, $\mathcal{C} = \mathcal{D}^{1/2}\mathcal{F}\mathcal{O}\mathcal{D}^{-1/2}\mathcal{B}$ est aussi solution puisque $\mathrm{Var}(\mathcal{C}\mathbf{x}) = \mathcal{D}^{1/2}\mathcal{F}\mathcal{O}\mathcal{D}^{-1/2}\mathrm{Var}(\mathcal{B}\mathbf{x})\mathcal{D}^{-1/2}\mathcal{O}^\mathsf{T}\mathcal{F}\mathcal{D}^{1/2} = \mathcal{D}^{1/2}\mathcal{F}\mathcal{O}\mathcal{D}^{-1/2}\mathcal{D}\mathcal{D}^{-1/2}\mathcal{O}^\mathsf{T}\mathcal{F}\mathcal{D}^{1/2} = \mathcal{D}^{1/2}\mathcal{F}\mathcal{O}\mathcal{O}^\mathsf{T}\mathcal{F}\mathcal{D}^{1/2} = \mathcal{D}\mathcal{F}^2$ qui est une matrice diagonale ayant tous ses éléments diagonaux positifs ou nuls.

On peut montrer que $\mathcal{B} = \mathcal{E}^\mathsf{T}$ est une solution au problème (P). En utilisant la décomposition en valeurs propres $\mathcal{E}\Lambda\mathcal{E}^\mathsf{T}$ de la matrice hermitienne $\mathrm{Var}(\mathbf{x})$, on obtient $\mathrm{Var}(\mathbf{Y}) = \mathcal{E}^\mathsf{T}\mathrm{Var}(\mathbf{x})\mathcal{E} = \mathcal{E}^\mathsf{T}\mathcal{E}\Lambda\mathcal{E}^\mathsf{T}\mathcal{E} = \Lambda$ et on retombe alors sur les résultats de la sous-section précédente.

La matrice de séparation $\mathcal{B} = \mathcal{E}^\mathsf{T}$ est orthogonale (c'est donc une rotation) mais d'autres choix sont possibles pour décorréler les observations. Ces autres choix de $\mathcal{B}$ ne mèneront pas à l'ACP et ne seront pas forcément des rotations. C'est donc le choix de maximiser les éléments diagonaux de $\text{Var}(\mathbf{Y})$ qui a mené à l'ACP. Notons tout de même que ce choix très particulier de $\mathcal{B}$ reste largement injustifié ici puisqu'il sous-entend que la matrice de mélange $\mathcal{A}$ est une rotation et rien ne laisse penser que nous avons le droit de nous limiter à ce type de transformations linéaires.

Remarque 3. *Nous verrons plus tard que la méthode d'Analyse en Composantes Indépendantes procède de manière tout à fait similaire puisqu'il s'agit aussi de trouver une matrice $\mathcal{B}$ de décorrélation mais où l'indétermination sera alors levée par la maximisation d'une quantité appelée néguentropie.*

Vision de l'ACPs par maximisation de l'inertie

Chacune des v_m colonnes correspond à un individu statistique. Chacune des t_m lignes correspond à une variable. On peut donc représenter le nuage des v_m individus dans un espace euclidien de dimension t_m noté $\mathbb{R}^{t_m}$. Nous nous intéressons à la proximité entre les individus dans ce nuage qui traduit toute l'information disponible. Il est alors possible de calculer une quantité numérique $\mathbf{I}$, appelée inertie, qui détermine la forme du nuage :

$$\mathbf{I} = \frac{1}{2v_m}\sum_{i=1}^{v_m}\sum_{j=1}^{v_m} d^2(\text{voxel}_i, \text{voxel}_j) = \frac{1}{2v_m}\sum_{i=1}^{v_m}\sum_{j=1}^{v_m} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{5.9}$$

L'inertie est donc la moyenne des carrés des inter-distances entre tous les voxels, chaque voxel étant caractérisé par un vecteur (de coordonnées) de taille t_m (son décours temporel), c'est-à-dire la valeur qu'il prend aux t_m temps. La distance d utilisée est la distance euclidienne.

On va alors chercher à projeter le nuage des v_m points individus sur un certain sous-espace, par exemple un plan, et choisir ce plan de telle façon que la déformation engendrée par la projection soit la plus faible possible (il sera appelé plan principal). Ainsi, l'information sera en grande partie conservée dans ce sous-espace. La déformation sera mesurée ici par la variation d'inertie après projection. Puisqu'une opération de projection raccourcit toujours les distances, il suffit de trouver le plan sur lequel le nuage projeté aura la plus grande inertie possible.

Il est possible de montrer que ce plan est caractérisé par les deux premiers vecteurs propres de la matrice de covariance et que l'on retombe donc sur les mêmes solutions que précédemment.

Pour illustrer de façon plus concrète ce qui précède, on peut se baser sur l'exemple du simulateur de signaux IRMF présenté dans les annexes en page 51. Un signal créneau en 0-1 (convolué avec la réponse hémodynamique) est présent dans les voxels d'une certaine région. Les voxels de cette région auront donc des valeurs fortes pour certains temps (quand le signal créneau vaut 1) et donc ces voxels seront isolés des autres dans $\mathbb{R}^{t_m}$ (et idéalement aussi sur le plan où on aura projeté).
L'ACPs permet donc d'étudier les voxels et de regrouper les voxels qui ont un "comportement" similaire, c'est-à-dire des valeurs proches pour tous les t_m temps. Les motifs spatiaux intéressants seront alors constitués des groupes de voxels qui se distinguent des autres.
La figure 5 présente le nuage des $v_m = 15 \times 15$ voxels de l'exemple de l'annexe D projetés sur le premier plan principal.

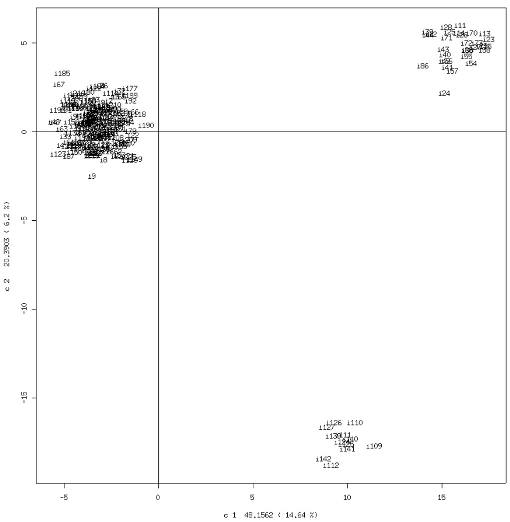


FIG. 5 – Nuage des v_m voxels de l'exemple de l'annexe D projetés sur le premier plan principal.

On peut nettement voir sur ce graphique trois groupes de voxels isolés les uns des autres. Le petit amas situé en bas est constitué par tous les voxels du patch rectangulaire de l'exemple (zone B), celui en haut à droite regroupe les voxels de la zone A dans laquelle est présent le signal créneau. Le reste des voxels sont regroupés au centre et ce sont ceux qui ne contiennent que du bruit.
Le premier axe $\boldsymbol{w}_1$ qui est égal à la première colonne de la matrice de mélange $\mathcal{A}$ (vecteur de coefficients, chaque coefficient étant associé à un temps) devrait donc avoir des coefficients forts pour les temps où le signal créneau vaut 1.

En effet, la valeur Y_i^1 de la première composante pour le voxel numéro i est, après avoir rajouté la moyenne, donnée par $Y_i^1 = \sum_{t=1}^{t_m} w_{1t} X_i^t$. Ainsi, lorsque Y_i^1 est fortement positif, alors X_i^t (qui est toujours positif) sera grand lorsque $w_{1t} > 0$ et petit lorsque $w_{1t} < 0$.

Liens entre l'ACPs et la DVS

Toute matrice rectangulaire peut être décomposée en utilisant la Décomposition en Valeurs Singulières (DVS). Ainsi la DVS de $\dot{X}$ s'écrit

$$\dot{X} = \mathcal{U}_f \mathcal{D}_f \mathcal{V}_f^{\mathsf{T}}. \tag{5.10}$$

La matrice $\mathcal{D}_f$ de taille $t_m \times t_m$ est une matrice diagonale dont les éléments diagonaux, ordonnés par valeurs décroissantes, sont appelés les valeurs singulières de $\dot{X}$. Les colonnes de la matrice $\mathcal{U}_f$ de taille $t_m \times t_m$ (resp. $\mathcal{V}_f$ de taille $v_m \times t_m$) sont les vecteurs singuliers à gauche (resp. à droite) associés à ces valeurs singulières.

On peut montrer qu'il y a une relation entre cette décomposition de $\dot{X}$ et la décomposition aux valeurs propres de $v_m S_{\mathrm{X}} = \dot{X}\dot{X}^{\mathsf{T}}$ suivante (valable uniquement pour les matrices carrées) :

$$v_m S_{\mathrm{X}} = \mathcal{E}_f \Lambda_f \mathcal{E}_f^{\mathsf{T}}. \tag{5.11}$$

En effet, on a

$$\Lambda_f = \mathcal{D}_f^2 \tag{5.12}$$

et

$$\mathcal{E}_f = \mathcal{U}_f \text{ (au signe près des colonnes).} \tag{5.13}$$

Il est aussi intéressant de noter que la DVS de $v_m S_{\mathrm{X}}$ est identique à la décomposition aux valeurs propres de $v_m S_{\mathrm{X}}$.

La décomposition en valeurs singulières compacte (*compact DVS*) de la matrice $\dot{X}$ s'écrit

$$\dot{X} = \mathcal{U}\mathcal{D}\mathcal{V}^{\mathsf{T}} = \sum_{k=1}^{t_m} d_k \boldsymbol{u}_k \otimes \boldsymbol{v}_k^{\mathsf{T}} \tag{5.14}$$

où l'on a posé $\mathcal{U} = [\boldsymbol{u}_1|\ldots|\boldsymbol{u}_{t_m}]$ et $\mathcal{V} = [\boldsymbol{v}_1|\ldots|\boldsymbol{v}_{t_m}]$.
Notons que l'on peut aussi approcher la matrice $\dot{X}$ en utilisant une DVS tronquée :

$$\dot{X} \approx \dot{X}_{red} = \mathcal{U}_{red}\mathcal{D}_{red}\mathcal{V}_{red}^{\mathsf{T}} = \sum_{k=1}^{r} d_k \boldsymbol{u}_k \otimes \boldsymbol{v}_k^{\mathsf{T}} \tag{5.15}$$

où l'on n'a utilisé que les r premières plus grandes valeurs singulières d_k. Cela peut être beaucoup plus rapide et économique en terme de stockage dans la mémoire que la DVS compacte.

Notons que chaque valeur propre d_k^2 spécifie le degré de variance dans les données associée avec la direction définie par un vecteur propre correspondant dans $\mathcal{U}$. On peut donc laisser tomber les vecteurs propres associés à des petites valeurs propres car elles ne correspondent qu'à des variations triviales dans les données.

Chacun des v_m vecteurs colonnes de dimension t_m dans $\dot{X}$ définit un seul point dans un espace de dimension t_m. Si la plupart de ces points vivent dans un sous-espace de dimension k (où $k << t_m$) alors on peut utiliser k vecteurs de bases $\mathcal{U}_{red}$ de dimension t_m judicieusement choisis pour représenter les v_m colonnes de $\dot{X}$ (par exemple si tous les points dans une boite se trouvent en fait sur un carré de dimension 2 alors on peut décrire les points en terme des deux vecteurs de base définis par les deux côtés de ce carré).

Choix du nombre de composantes

Une autre question intéressante est celle du nombre q de composantes principales à retenir pour représenter de façon satisfaisante les données. Pour cela, on peut se baser sur le critère empirique du pourcentage de la variation totale expliquée par les q premières composantes :

$$100\frac{\lambda_1 + \ldots + \lambda_q}{\lambda_1 + \ldots + \lambda_{t_m}}. \tag{5.16}$$

On trace aussi souvent l'éboulis (spectre) des valeurs propres rangées par ordre décroissant pour y détecter une rupture.

5.2 L'Analyse en Composantes Principales temporelle (ACPt)

Comme en ACP spatiale (ACPs), il y a trois façons de voir l'ACP temporelle (notée ACPt dans la suite) :

1. Par maximisation de la variance,
2. Par recherche de sources non corrélées,
3. Par maximisation de l'inertie.

Nous ne les détaillerons pas ici.

On considère que l'on a un vecteur aléatoire $\mathbf{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_{v_m})^{\mathsf{T}}$ à v_m composantes et que l'on dispose d'un échantillon de taille t_m de tels vecteurs : $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{t_m}$. On peut donc regrouper les données IRMF dans une matrice de dimension $v_m \times t_m$ qui est la transposée de celle utilisée dans le cas de l'ACPs et sera notée X^{T}.

$$X^{\mathsf{T}} = [\boldsymbol{x}^1|\ldots|\boldsymbol{x}^{t_m}] = \begin{pmatrix} X_1^1 & \ldots & X_1^t & \ldots & X_1^{t_m} \\ \vdots & & \vdots & & \vdots \\ X_{v_m}^1 & \ldots & X_{v_m}^t & \ldots & X_{v_m}^{t_m} \end{pmatrix} = \begin{pmatrix} \text{Décours temporel } 1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \text{Décours temporel } v_m \end{pmatrix}. \tag{5.17}$$

La colonne numéro t de $X^{\intercal}$ représente les données dont on dispose au temps t pour l'ensemble des v_m voxels. L'objectif de l'ACPt est de rechercher des décours temporels intéressants dans les données IRMF.

La méthodologie employée dans l'ACP temporelle est la même que celle de l'ACP spatiale mais appliquée sur la matrice des données transposée.
La recherche des composantes principales se fait donc en calculant les valeurs et vecteurs propres de la matrice de covariance $S_{X^{\intercal}} = \frac{1}{t_m}\dot{X}^{\intercal}\dot{X}$.

Remarque 4. Notons que dans le cas de données IRMF, la matrice $S_{X^{\intercal}}$, qui est de taille $v_m \times v_m$, est difficile à diagonaliser puisque v_m est de l'ordre du million.

Toutefois, seuls les premiers vecteurs propres (associés aux plus grandes valeurs propres) nous intéressent puisque ce sont eux qui apportent le plus d'information et il est possible d'utiliser des méthodes itératives (voir [WRW+05] et [AS07]) pour les obtenir.

Une autre approche est aussi possible qui permet d'obtenir les t_m premiers vecteurs propres du cas temporel à partir de ceux du cas spatial (voir [AGA99]).

Le lecteur pourra consulter le Résultat 1 des annexes en page 44 pour plus de détails.

Le vecteur $\boldsymbol{Y}$ des t_m **premières** composantes principales du cas temporel est donné par :

$$\boldsymbol{Y} = \mathcal{F}^{\intercal}\boldsymbol{X} \tag{5.18}$$

où $\mathcal{F} = [\boldsymbol{f}_1|\ldots|\boldsymbol{f}_{t_m}]$ est la sous-matrice contenant les t_m **premiers** vecteurs propres de $t_m S_{X^{\intercal}}$ en colonnes.

La matrice $\mathcal{F}$ des vecteurs propres de $S_{X^{\intercal}}$ est la matrice de la nouvelle base (exprimée dans la base canonique) dans laquelle on va "regarder" le nuage. En particulier, les deux premières colonnes de cette matrice (la première colonne est le vecteur propre associé à la plus grande valeur propre, la deuxième colonne est le vecteur propre associé à la deuxième plus grande valeur propre, etc ...) sont les vecteurs qui déterminent (dans la base canonique) les deux axes du plan principal. Ces vecteurs propres sont appelés les axes (ou facteurs) principaux de l'ACPt.

On notera $\mathcal{C}_{\mathcal{F}}$ la matrice des composantes principales (coordonnées des points individus dans la nouvelle base des vecteurs propres $\mathcal{F}$) :

$$\mathcal{C}_{\mathcal{F}} = \mathcal{F}^{\intercal}X^{\intercal}. \tag{5.19}$$

Les coordonnées de l'individu t (temps t) sont dans la t-ème colonne de cette matrice.

Liens entre l'ACPt et la DVS

Toute matrice peut être décomposée en utilisant la Décomposition en Valeurs Singulières (DVS). Ainsi la DVS de $\dot{X}^{\intercal}$ s'écrit

$$\dot{X}^{\intercal} = \mathcal{V}_f\mathcal{D}_f\mathcal{U}_f^{\intercal}. \tag{5.20}$$

La matrice $\mathcal{D}_f$ de taille $v_m \times v_m$ est une matrice diagonale dont les éléments diagonaux, ordonnés par valeurs décroissantes, sont appelés les valeurs singulières de $\dot{X}^{\intercal}$. Les colonnes de la matrice $\mathcal{V}_f$ de taille $v_m \times v_m$ (resp. $\mathcal{U}_f$ de taille $t_m \times v_m$) sont les vecteurs singuliers à gauche (resp. à droite) associés à ces valeurs singulières.

Dans le cas qui nous intéresse ici ($t_m \leq v_m$), on peut montrer que les $v_m - t_m$ dernières valeurs singulières de $\mathcal{D}_f$ sont nulles et l'on peut alors réécrire (5.20) sous une forme plus compacte appelée la DVS compacte de $\dot{X}^{\intercal}$:

$$\dot{X}^{\intercal} = \mathcal{V}\mathcal{D}\mathcal{U}^{\intercal}, \tag{5.21}$$

où ici on retrouve les matrices de la DVS de $\dot{X}$: $\mathcal{D}$ est de taille $t_m \times t_m$, $\mathcal{V}$ est de taille $v_m \times t_m$ et $\mathcal{U}$ est de taille $t_m \times t_m$.

On peut montrer qu'il y a une relation entre la DVS de $\dot{X}^\intercal$ et la décomposition aux valeurs propres de $t_m S_{\mathrm{X}^\intercal} = \dot{X}^\intercal \dot{X}$ suivante (valable uniquement pour les matrices carrées) :

$$t_m S_{\mathrm{X}^\intercal} = \mathcal{F}_f \Lambda_f \mathcal{F}_f^\intercal = \mathcal{F} \Lambda \mathcal{F}^\intercal, \tag{5.22}$$

où Λ_f est la matrice diagonale de taille $v_m \times v_m$ des valeurs propres de $t_m S_{\mathrm{X}^\intercal}$ et $\mathcal{F}_f$ est la matrice de taille $v_m \times v_m$ des vecteurs propres associés. L'écriture du membre de droite de l'équation précédente vient du fait que les $v_m - t_m$ dernières valeurs propres de Λ_f sont nulles. On note alors Λ, de taille $t_m \times t_m$, et $\mathcal{F}$, de taille $v_m \times t_m$, les sous-matrices correspondantes.
On a les relations suivantes

$$\Lambda_f = \mathcal{D}_f^2 \tag{5.23}$$

et

$$\mathcal{F}_f = \mathcal{V}_f \text{ (au signe près des colonnes).} \tag{5.24}$$

On a aussi bien entendu

$$\Lambda = \mathcal{D}^2 \tag{5.25}$$

et

$$\mathcal{F} = \mathcal{V} \text{ (au signe près des colonnes).} \tag{5.26}$$

Il est aussi intéressant de noter que la DVS de $t_m S_{\mathrm{X}^\intercal}$ est identique à la décomposition aux valeurs propres de $t_m S_{\mathrm{X}^\intercal}$.

La décomposition en valeurs singulières compacte de la matrice $\dot{X}^\intercal$ s'écrit

$$\dot{X}^\intercal = \mathcal{V}\mathcal{D}\mathcal{U}^\intercal = \sum_{k=1}^{t_m} d_k \boldsymbol{v}_k \otimes \boldsymbol{u}_k^\intercal \tag{5.27}$$

où l'on a posé $\mathcal{V} = [\boldsymbol{v}_1|\dots|\boldsymbol{v}_{t_m}]$ et $\mathcal{U} = [\boldsymbol{u}_1|\dots|\boldsymbol{u}_{t_m}]$.
Notons que l'on peut aussi approcher la matrice $\dot{X}^\intercal$ en utilisant une DVS tronquée :

$$\dot{X}^\intercal \approx \dot{X}_{red}^\intercal = \mathcal{V}_{red}\mathcal{D}_{red}\mathcal{U}_{red}^\intercal = \sum_{k=1}^{r} d_k \boldsymbol{v}_k \otimes \boldsymbol{u}_k^\intercal \tag{5.28}$$

où l'on n'a utilisé que les r premières plus grandes valeurs singulières. Cela peut être beaucoup plus rapide et économique en terme de stockage dans la mémoire que la DVS compacte.

Notons que chaque valeur propre d_k^2 spécifie le degré de variance dans les données associé avec la direction définie par un vecteur propre correspondant dans $\mathcal{V}$. On peut donc laisser tomber les vecteurs propres associés à des petites valeurs propres car elles ne correspondent qu'à des variations triviales dans les données.

Chacun des t_m vecteurs colonnes de dimension v_m dans $\dot{X}^\intercal$ définit un seul point dans un espace de dimension v_m. Si la plupart de ces points vivent dans un sous-espace de dimension k (où $k << v_m$) alors on peut utiliser k vecteurs de bases $\mathcal{V}_{red}$ de dimension v_m judicieusement choisis pour représenter les t_m colonnes de $\dot{X}^\intercal$ (par exemple si tous les points dans une boite 3D se trouvent en fait sur un carré de dimension 2 alors on peut décrire ces points en terme des deux vecteurs de base définis par les deux côtés de ce carré).

Par ailleurs, si l'on pose, $\mathcal{A} = \mathcal{V}$ et $S = \mathcal{D}\mathcal{U}^{\mathsf{T}}$, alors $\dot{X}^{\mathsf{T}} = \mathcal{A}S$.
On peut alors interpréter la DVS comme une estimation du modèle à variables latentes $\mathbf{x} = \mathcal{A}\mathbf{s}$ suivant :

$$\begin{array}{rcl} \mathrm{x}_1 & = & a_{11}\mathrm{s}_1 + a_{12}\mathrm{s}_2 + \ldots + a_{1t_m}\mathrm{s}_{t_m} \\ \mathrm{x}_2 & = & a_{21}\mathrm{s}_1 + a_{22}\mathrm{s}_2 + \ldots + a_{2t_m}\mathrm{s}_{t_m} \\ \vdots & = & \vdots \\ \mathrm{x}_{v_m} & = & a_{v_m1}\mathrm{s}_1 + a_{v_m2}\mathrm{s}_2 + \ldots + a_{v_mt_m}\mathrm{s}_{t_m}. \end{array}$$

Les variables corrélées x_j sont représentées comme des combinaisons linéaires des variables non corrélées s_l de variance unité. Cela n'est cependant pas tout à fait satisfaisant car on peut écrire, pour toute matrice orthogonale $\mathcal{O}$:

$$\mathbf{x} = \mathcal{A}\mathbf{s} = \mathcal{A}\mathcal{O}^{\mathsf{T}}\mathcal{O}\mathbf{s} = \mathcal{A}^*\mathbf{s}^* \tag{5.29}$$

et $\mathrm{Var}(\mathbf{s}^*) = \mathcal{O}\mathrm{Var}(\mathbf{s})\mathcal{O}^{\mathsf{T}} = \mathcal{I}$. Par conséquent, il y a plusieurs décompositions de ce type et il n'est donc pas possible d'identifier un jeu particulier de variables latentes comme les sources uniques sous-jacentes.
On voit que l'on retombe ici sur la vision de l'ACP par recherche de sources non corrélées.

6 Le problème de la séparation de sources

6.1 Heuristique et méthodes de résolution

La plupart des quantités physiques mesurées sont constituées d'un mélange d'autres quantités. Des exemples typiques sont : i) des signaux sonores dans une pièce où plusieurs personnes discutent simultanément, ii) un signal EEG qui contient des contributions de plusieurs régions cérébrales différentes et iii) la taille d'une personne qui est déterminée par les contributions de nombreux facteurs génétiques et environnementaux. Sous certaines conditions, il est possible de retrouver les composantes (sources) sous-jacentes des quantités mesurées en utilisant la méthode d'Analyse en Composantes Indépendantes (ACI) ou la méthode de poursuite de projection (PP).

L'hypothèse sur laquelle repose la première méthode (ACI) de recherche de ces sources est que les différents processus physiques tendant à générer des signaux sont statistiquement indépendants les uns des autres. Cela suggère qu'un moyen de retrouver les signaux sources à partir de l'observation d'un mélange de ces signaux est de trouver des transformations de ces mélanges qui produisent des composantes (signaux extraits) indépendantes. D'un autre côté, la Poursuite de Projection (PP), une autre méthode de recherche des sources, repose sur l'hypothèse que toute combinaison linéaire d'un ensemble de signaux sources (de variance finie) est gaussienne et que les signaux sources eux-mêmes sont non gaussiens. Ainsi, une autre méthode pour extraire des signaux sources à partir d'un mélange de ces signaux est de trouver des transformations de ces mélanges qui extraient des composantes non gaussiennes. On peut montrer que l'hypothèse d'indépendance statistique est implicite dans l'hypothèse qu'au plus un signal source est gaussien, et par conséquent que les méthodes PP et ACI sont basées sur les mêmes hypothèses.

6.2 Principes

Soit n_m signaux sources variant au cours du temps. On définit les amplitudes de ces signaux au temps t par un vecteur colonne $\boldsymbol{s} = (s_1^t, \ldots, s_{n_m}^t)^{\mathsf{T}}$. Ces signaux peuvent être combinés de façon linéaire pour former un signal mélange $x = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{s}$, où chaque élément du vecteur ligne $\boldsymbol{a}^{\mathsf{T}}$ spécifie de combien le signal source correspondant s_i^t contribue au signal mélange x. Étant donné v_m signaux mélanges $\boldsymbol{x} = (x_1^t, \ldots, x_{v_m}^t)^{\mathsf{T}}$ on peut définir une matrice de mélange $\mathcal{A} = [\boldsymbol{a}_1^{\mathsf{T}}; \ldots; \boldsymbol{a}_{v_m}^{\mathsf{T}}]$ de taille $v_m \times n_m$ dans laquelle chaque ligne $\boldsymbol{a}_i^{\mathsf{T}}$ spécifie un unique mélange x_i^t des signaux $\boldsymbol{s} = (s_1^t, \ldots, s_{n_m}^t)^\mathsf{T}$. En utilisant cette notation matricielle, la formation de v_m signaux mélanges à partir de n_m signaux sources peut s'écrire

$$\boldsymbol{x} = \mathcal{A}\boldsymbol{s}. \tag{6.1}$$

L'ACI et la PP sont capables de retrouver les sources $\boldsymbol{s}$ à partir des signaux mélanges $\boldsymbol{x}$. Le fait que les signaux mélanges puissent être séparés *en principe* est maintenant facile à démontrer. Une matrice de séparation $\mathcal{W}$ est définie par

$$\boldsymbol{s} = \mathcal{W}\boldsymbol{x}. \tag{6.2}$$

Étant donné que chaque ligne de $\mathcal{W}$ spécifie comment les mélanges dans $\boldsymbol{x}$ sont recombinés pour produire un signal source, il s'ensuit qu'il doit être possible de retrouver un signal à la fois, en utilisant différents vecteurs lignes pour extraire chaque signal. Par exemple, si un seul signal source doit être extrait de v_m signaux mélanges alors $\mathcal{W}$ est une matrice de taille $1 \times v_m$. Ainsi, la forme de la matrice de séparation $\mathcal{W}$ dépend du nombre de signaux qui doivent être extraits. Habituellement, l'ACI est utilisée pour extraire plusieurs signaux sources simultanément alors que la PP est utilisée pour extraire un seul signal source à la fois.

Il est toutefois bon de noter que certaines indéterminations seront forcément présentes dans la résolution de l'équation 6.1 :

– On ne peut pas déterminer les variances (énergies) des composantes indépendantes. L'explication est que, étant donné que $\mathbf{s}$ et $\mathcal{A}$ sont toutes les deux inconnues, si l'on multiplie l'une des sources s_i par un scalaire α_i il suffit de diviser la colonne correspondante $\boldsymbol{a}_i$ par α_i et l'on ne change rien au problème :

$$\mathbf{x} = \sum_i \left(\frac{1}{\alpha_i}\boldsymbol{a}_i\right)(\mathrm{s}_i\alpha_i). \tag{6.3}$$

  Par conséquent, sans perte de généralité, nous imposerons la restriction $\mathbf{E}[\mathrm{s}_i] = 1$. Cela laisse toujours l'ambiguité du signe, ce qui n'est pas gênant dans nos applications.

– On ne peut pas déterminer l'ordre des composantes indépendantes. Ainsi, pour toute matrice de permutation $\mathcal{P}$, on a $\mathbf{x} = \mathcal{A}\mathbf{s} = \mathcal{A}\mathcal{P}^{-1}\mathcal{P}\mathbf{s}$, les élements de $\mathcal{P}\mathbf{s}$ étant les sources indépendantes originales s_j mais présentées dans un ordre différent.

Remarque 5. *À ce stade, nous pouvons donner un exemple des sources potentielles en IRMF : les battements cardiaques, le cycle respiratoire, les mouvements occulaires, les imperfections de la machine IRM (entrainant des dérives du signal), et le stimulus de l'expérience !*

6.3 La géométrie de la séparation de sources

La nature géométrique de la matrice $\mathcal{W}$ de séparation peut être explorée en considérant le cas de deux signaux mélanges. Considérons deux signaux $\boldsymbol{s}^t = \{s_1^t, s_2^t\}$ qui ont été mélangés au moyen d'une matrice $\mathcal{A}$ de taille 2×2 pour produire deux mélanges de signaux $\boldsymbol{x}^t = \mathcal{A}\boldsymbol{s}^t$ avec $\boldsymbol{x}^t = \{x_1^t, x_2^t\}$. Dans l'exemple de la figure 6, chacune des sources est constituée de deux portions de gaussiennes indépendantes temporellement de moyenne 0 et d'écart-type 1.5 et 0.07 pour chacune des portions respectivement. La matrice de mélange $\mathcal{A} = \mathcal{A}_1\mathcal{A}2$ est la composition des deux matrices de gauchissement (*shear*) $\mathcal{A}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ et $\mathcal{A}_2 = \begin{pmatrix} 1 & 0.5 \\ 0 & 1 \end{pmatrix}$.

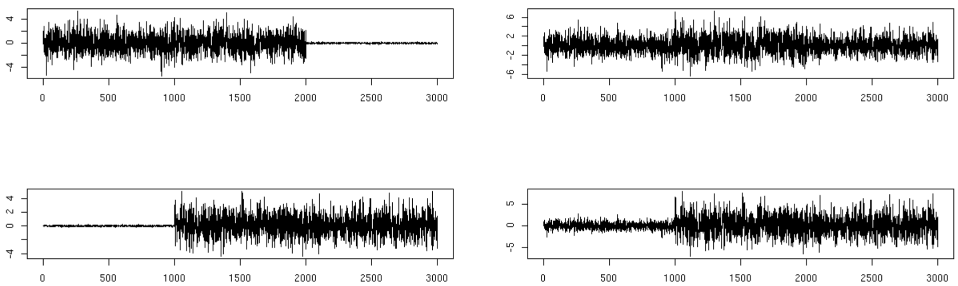


FIG. 6 – À gauche : les signaux sources (s_1^t, s_2^t) . À droite : les mélanges (x_1^t, x_2^t) obtenus à partir d'une matrice de mélange composée de deux matrices de gauchissement respectivement parallèle à l'axe des abscisses de paramètre 1, et parallèle à l'axe des ordonnées de paramètre 0.5.

Si on interprète ceci en terme de deux voix humaines (sources) et deux microphones (mélanges), alors les éléments de la i-ème ligne de $\mathcal{A}$ spécifient la proximité de chaque voix au i-ème microphone. Chaque microphone enregistre un mélange pondéré x_i^t des deux sources s_1^t et s_2^t, où les poids pour chaque microphone sont donnés par une ligne de $\mathcal{A}$.
Un graphique de s_1^t versus s_2^t, et de x_1^t versus x_2^t est présenté sur la figure 7.

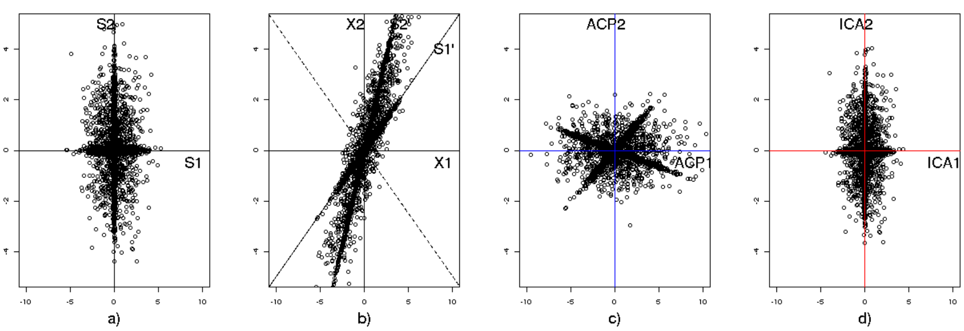


FIG. 7 – a) Les sources (s_1^t, s_2^t). b) Les mélanges (x_1^t, x_2^t). c) Les axes en bleu représentent les directions orthogonales d'allongement maximal de la distribution obtenues par l'ACP. d) Les axes rouges sont obtenus par l'ACI.

On note $\mathbf{S}$ l'espace dans lequel vit $\boldsymbol{s}^t$ muni des axes S_1 et S_2, et on note $\mathbf{X}$ l'espace dans lequel vit $\boldsymbol{x}^t$ muni des axes X_1 et X_2. Les amplitudes des signaux sources s_1^t et s_2^t au temps t $(1 \leq t \leq t_m)$ sont représentées par le point $\boldsymbol{s}^t$ de coordonnées $(s_1^t, s_2^t)^\mathsf{T}$ dans $\mathbf{S}$. Les amplitudes correspondantes des mélanges de signaux $\boldsymbol{x}^t = \mathcal{A}\boldsymbol{s}^t$ au temps t sont représentées par un point $\boldsymbol{x}^t$ de coordonnées $(x_1^t, x_2^t)^\mathsf{T}$ dans $\mathbf{X}$. La matrice de mélange $\mathcal{A}$ définit une transformation linéaire (dans laquelle les lignes de $\mathcal{A}$ forment les vecteurs de base dans $\mathbf{X}$), si bien que le passage de $\mathbf{S}$ à $\mathbf{X}$ consiste en une rotation et un gauchissement (*shear matrix*) des axes dans $\mathbf{S}$. Ainsi, les axes orthogonaux S_1 et S_2 dans $\mathbf{S}$ apparaissent comme deux lignes obliques S_1' et S_2' dans $\mathbf{X}$. Notons que la variation le long de chaque axe dans $\mathbf{S}$ est causée par la variation en amplitude d'un signal source. Étant donné que chaque axe S_i dans $\mathbf{S}$ correspond à une direction S_i' dans $\mathbf{X}$, la variation le long des axes projetés S_1' et S_2' dans $\mathbf{X}$ sont causées par la variation des amplitudes des signaux définis pas s_1^t et s_2^t respectivement. Si l'on peut extraire les variations associées avec une direction, par exemple S_1'

dans $\mathbf{X}$ en ignorant les variations suivant toutes les autres directions, alors on peut retrouver les amplitudes du signal s_1^t. Ceci peut être obtenu en projetant tous les points dans $\mathbf{X}$ sur une ligne qui est orthogonale à toutes les directions sauf une S_1'. Une telle ligne est définie par le vecteur $\boldsymbol{w}_1 = (w_1, w_2)^\mathsf{T}$ (ligne en pointillés sur la figure), définie de telle façon que seules les composantes de $\mathbf{X}$ qui sont suivant la direction S_1' sont transformées en des valeurs non nulles de $\boldsymbol{y}^t = \boldsymbol{w}_1 \boldsymbol{x}^t$.
Pour résumer, la transformation linéaire $y^t = \boldsymbol{w}^\mathsf{T} \boldsymbol{x}^t$ produit une valeur scalaire pour chaque point $\boldsymbol{x}^t$ dans $\mathbf{X}$, de telle façon qu'un seul signal résulte de la transformation $y^t = \boldsymbol{w}^\mathsf{T} \boldsymbol{x}^t$. L'amplitude du signal y^t au temps t est trouvée en prenant le produit scalaire de $\boldsymbol{w}$ avec un point $\boldsymbol{x}^t$. Comme le vecteur ligne $\boldsymbol{w}^\mathsf{T}$ est défini pour être orthogonal à toutes les directions correspondant aux signaux source sauf une dans $\mathbf{X}$, seulement ce signal sera projeté en des valeurs non nulles $y^t = \boldsymbol{w}^\mathsf{T} \boldsymbol{x}^t$.

7 Analyse par la méthode des Composantes Indépendantes (ACI)

Les méthodes d'ACI décrites dans cette section sont basées sur l'observation simple suivante. Si un ensemble de n_m signaux proviennent de différentes sources physiques (par exemple n_m personnes parmi les invités d'un cocktail, ne se parlant pas entre elles et ne parlant pas de la même chose) alors ces signaux tendent à être statistiquement indépendants entre eux. La méthode d'ACI est basée sur l'hypothèse que si un ensemble de signaux indépendants peuvent être extraits à partir d'un mélange de signaux alors ces signaux extraits ont de grandes chances d'être les signaux sources originaux.
Il existe diverses mesures de la dépendance entre ces signaux qui mènent à différents algorithmes. Nous en présentons quelques unes dans la sous-section 7.1. Comme dans le cas de l'ACP, on pourra considérer une ACI spatiale et une ACI temporelle. Les sous-sections 7.2 et 7.3 décrivent ces deux approches respectivement.

7.1 La mise en oeuvre pratique de l'ACI

Un très grand nombre de méthodes pratiques ont été développées ces dernières années pour estimer le modèle d'ACI, c'est-à-dire résoudre le problème de la séparation des sources dans (5.8) mais avec l'hypothèse plus forte que les sources sont indépendantes. Chacune de ces méthodes est basée sur un principe différent et donne lieu à un algorithme particulier. Nous pouvons par exemple citer les méthodes basées sur l'estimation du maximum de vraisemblance, celles basées sur la minimisation de l'information mutuelle et celles basées sur la maximisation de la non gaussianité (équivalent à la PP). Nous décrivons ici succintement une méthode basée sur la maximisation de la Néguentropie, indice mesurant la non-gaussianité et une autre méthode basée sur la minimisation de l'information mutuelle.

– *Méthode utilisant la néguentropie*
  Pour estimer l'une des composantes source s_i, on considère une combinaison linéaire $\mathrm{Y} = \boldsymbol{b}^\mathsf{T}\mathbf{x} = \sum_i b_i \mathrm{x}_i$ des variables à observer x_i. Puisque $\mathbf{x} = \mathcal{A}\mathbf{s}$, on peut aussi écrire $\mathrm{Y} = \boldsymbol{q}^\mathsf{T}\mathbf{s} = \sum_i q_i \mathrm{s}_i$ avec $\boldsymbol{q} = \mathcal{A}^\mathsf{T}\boldsymbol{b}$. Intuitivement, par le théorème central limite, toute combinaison linéaire de variables possède une distribution plus gaussienne que chacune des variables d'origine. Ainsi, $\mathrm{Y} = \sum_i q_i \mathrm{s}_i$ sera la moins gaussienne possible lorsque tous les q_i seront nuls sauf un (disons q_{i_0}), auquel cas on aura $\mathrm{Y} = \mathrm{s}_{i_0}$ et l'on aura retrouvé la i_0-ème source ! Donc tout ce qu'il y à a faire c'est de trouver une variable aléatoire Y (ce qui est équivalent à trouver un vecteur $\boldsymbol{b}$) telle que Y soit la moins gaussienne possible.
  Nous avons donc besoin d'une mesure de non gaussianité d'une variable aléatoire : la néguentropie vue en (4.4) nous en fournit une. En pratique, on peut estimer la néguentropie $\mathbf{J}(\mathrm{Y})$ par

$$(\mathbf{E}[G(\mathrm{Y})] - \mathbf{E}[G(\nu)])^2 \tag{7.1}$$

  où G est une fonction non quadratique (par exemple $-\exp(-y^2/2)$) et $\nu \sim N(0,1)$.

– *Méthode utilisant l'information mutuelle*
  On a déjà vu que l'information mutuelle est une mesure naturelle de la dépendance entre des variables

aléatoires. Elle est toujours positive et elle vaut zéro uniquement si les variables sont statistiquement indépendantes. Il suffira donc de chercher une matrice $\hat{\mathcal{W}}$ telle que les sources estimées $\mathbf{Y} = \hat{\mathbf{s}} = \hat{\mathcal{W}}\mathbf{X}$ aient une information mutuelle $\mathbf{Inf}(\mathbf{Y}) = \mathbf{Inf}(\mathrm{Y}_1, \ldots, \mathrm{Y}_n)$ la plus faible possible.

Remarque 6. *Il est possible de montrer que* $\mathbf{Inf}(\mathrm{Y}_1, \ldots, \mathrm{Y}_n) = constante - \sum_i \mathbf{J}(\mathrm{Y}_i)$ *ce qui montre la relation fondamentale unissant la néguentropie et l'information mutuelle.*

Remarque 7. *La néguentropie permet l'estimation des composantes indépendantes l'une après l'autre (deflationary approach) alors que l'information mutuelle conduit à trouver l'ensemble des composantes simultanément (parallel approach).*

7.2 L'Analyse en Composantes Indépendantes spatiale (ACIs)

Normalement, des images spatialement indépendantes (le plus indépendantes possible en fait) sont extraites en plaçant chaque image observée (centrée) dans une ligne de $\dot{X}$, matrice de taille $t_m \times v_m$. Ceci traite essentiellement chacun des t_m pas de temps comme un microphone ou mélange d'images indépendantes, si bien que chaque mélange consiste en v_m voxels. Dans ce cas, chaque signal source estimé (ligne de $\hat{S}$) est une image, où les valeurs des pixels dans chaque image (ligne) sont indépendantes de toute autre image, si bien que ces images sont dites spatialement indépendantes. Chaque colonne de la matrice de mélange estimée $\hat{\mathcal{A}}$ de taille $t_m \times t_m$ est une séquence temporelle.

7.3 L'Analyse en Composantes Indépendantes temporelle (ACIt)

Au lieu de placer chaque image dans une ligne de $\dot{X}$, on peut placer l'image correspondant à chaque pas temporel dans une colonne de $\dot{X}$. Ceci traite essentiellement chacun des v_m voxels comme un microphone ou mélange distinct, si bien que chaque mélange consiste en t_m pas de temps. La grande $(v_m \times v_m)$ matrice $\hat{\mathcal{W}}$ de séparation trouve alors des sources temporellement indépendantes qui contribuent à chaque niveau de gris d'un pixel au cours du temps. Ayant découvert les signaux temporellement indépendants pour une suite d'images, cela pose la question : qu'est-ce qui a varié indépendamment au cours du temps ? Étant donné $\hat{S} = \hat{\mathcal{W}}\dot{X}$ on peut dériver

$$\dot{X} = \hat{\mathcal{A}}\hat{S} \tag{7.2}$$

où $\hat{\mathcal{A}} = \hat{\mathcal{W}}^{-1}$ est une matrice de taille $v_m \times v_m$. Ainsi, chaque ligne (signal source estimé) de $\hat{S}$ spécifie comment la contribution à $\dot{X}$ d'une colonne (image) de $\hat{\mathcal{A}}$ varie au cours du temps. Ainsi, alors que chaque ligne $\hat{\boldsymbol{s}}_i^{\mathsf{T}}$ de $\hat{S}$ spécifie un signal qui est indépendant de toutes les lignes dans $\hat{S}$, chaque colonne $\hat{\boldsymbol{a}}_i$ de $\hat{\mathcal{A}}$ consiste en une image qui varie indépendamment au cours du temps selon l'amplitude de $\hat{s}_i$. Notons que, en général, les lignes de $\hat{S}$ sont contraintes à être mutuellement indépendantes, alors que la relation entre les colonnes de $\hat{\mathcal{A}}$ est complètement libre.

En résumé, l'ACIs produit un ensemble d'images mutuellement aussi indépendantes que possible et un ensemble associé de décours temporels non contraints alors que l'ACIt produit un ensemble de décours temporels mutuellement aussi indépendants que possible et un ensemble correspondant d'images non contraintes. La figure 8 illustre les différences entre les deux approches.

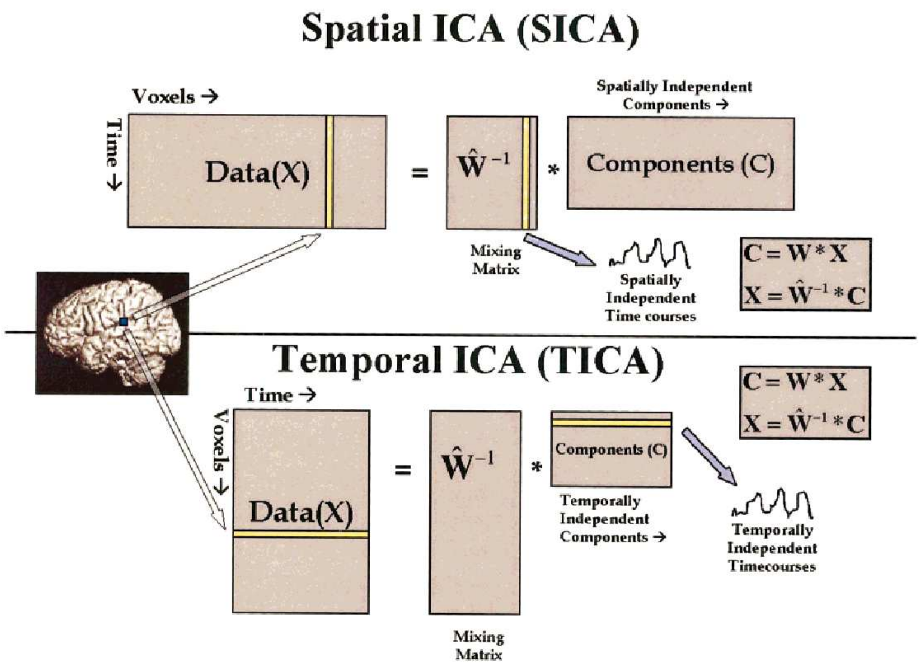


FIG. 8 – Description visuelle de l'ACIs versus l'ACIt. Dans chaque approche, les informations spatiale et temporelle sont déroulées dans une dimension. En ACIs, l'algorithme cherche à trouver des composantes spatialement indépendantes avec des décours temporels associés (non contraints) alors qu'en ACIt, l'algorithme cherche à trouver des décours temporels indépendants avec des cartes spatiales associées. D'après [CAPP01b].

Remarque 8. *S'il est connu que soit l'indépendance spatiale, soit l'indépendance temporelle ne peut être supposée, alors cela remet en question l'utilisation de l'ACIt ou de l'ACIs respectivement. En pratique, l'ACIt est une méthode demandant beaucoup de ressources informatiques calculatoires car elle implique une matrice* $\hat{\mathcal{W}}$ *de taille* $v_m \times v_m$*. A l'inverse, l'ACIs nécessite une matrice* $\hat{\mathcal{W}}$ *de taile* $t_m \times t_m$*.*

7.4 Utiliser l'ACP pour pré-traiter les données

Une étape préliminaire lors de l'utilisation des algorithmes d'ACI spatiale (resp. temporelle) consiste à effectuer un blanchiment (*whitening* ou *sphering*) des observations au moyen d'une ACP spatiale (resp. temporelle). Cette approche présente deux avantages : elle permet d'une part de réduire la dimension des observations et de limiter l'utilisation des ressources mémoire de l'ordinateur, et d'autre part elle représente une première étape de traitement permettant d'obtenir des signaux non corrélés en utilisant des statistiques du deuxième ordre. Il ne reste alors plus qu'à estimer une rotation qui minimise les dépendance aux ordres supérieurs.

On a vu que le problème de l'ACI n'est résolvable qu'à une matrice diagonale près. Nous supposerons donc, sans perte de généralité, que les sources sont de variance unité.
Ainsi $\mathrm{Var}(\mathbf{s}) = \mathcal{I}$. Si l'on a trouvé une matrice de blanchiment $\mathcal{B}$ telle que $\mathbf{z} = \mathcal{B}\mathbf{x}$ et $\mathrm{Var}(\mathbf{z}) = \mathcal{I}$ (par exemple en utilisant l'ACP), on peut essayer de trouver une matrice $\hat{\mathcal{W}}_\mathrm{X} = \hat{\mathcal{W}}_\mathrm{Z}\mathcal{B}$ de séparation telle que les sources estimées $\hat{\mathbf{s}} = \hat{\mathcal{W}}_\mathrm{X}\mathbf{x} = \hat{\mathcal{W}}_\mathrm{Z}\mathbf{z}$ soient (le plus possible) indépendantes. L'indépendance impliquant la non corrélation, on peut exiger que les sources estimées $\hat{\mathbf{s}}$ soient non corrélées, i.e. $\mathrm{Var}(\hat{\mathbf{s}}) = \mathcal{I}$. De ceci, il découle que l'on doit avoir $\mathrm{Var}(\hat{\mathcal{W}}_\mathrm{Z}\mathbf{z}) = \hat{\mathcal{W}}_\mathrm{Z}\mathrm{Var}(\mathbf{z})\hat{\mathcal{W}}_\mathrm{Z}^\intercal = \mathcal{I}$ et donc que $\hat{\mathcal{W}}_\mathrm{Z}\hat{\mathcal{W}}_\mathrm{Z}^\intercal = \mathcal{I}$, i.e. que $\hat{\mathcal{W}}_\mathrm{Z}$ est orthogonale (c'est-à-dire une rotation).

Ainsi, après avoir blanchi les observations, il ne reste plus qu'une rotation à trouver pour obtenir des signaux (le plus possible) indépendants.
Le gain est important puisqu'on vient ainsi de passer de l'estimation de n^2 termes pour une matrice quelconque à $n(n-1)/2$ termes pour une rotation.

Le diagramme suivant décrit la procédure à utiliser

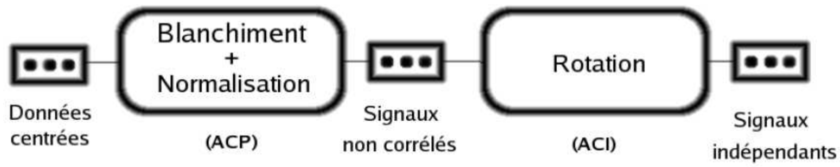


FIG. 9 – Un blanchiment préalable des données ne laisse plus qu'une rotation à estimer par l'algorithme d'ACI.

alors que le graphique suivant l'illustre sur deux variables sources indépendantes de loi jointe uniforme sur un carré.

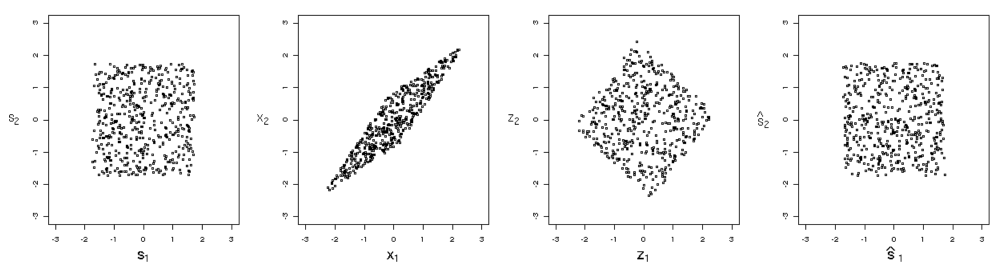


FIG. 10 – De gauche à droite : les sources originales de loi jointe uniforme sur un carré, les signaux mélangés, les composantes décorrélées obtenues par l'ACP, les sources estimées par l'ACI.

7.5 Appliquer une ACIs après une ACPs

L'idée est donc de commencer par effectuer une ACPs de $\dot{X}$, pour utiliser une ACIs non pas sur $\dot{X}$ mais sur la matrice des composantes principales (ou plutôt une version normalisée de celle-ci).

Posons $Y = \mathcal{E}^\mathsf{T}\dot{X} = \mathcal{U}^\mathsf{T}\dot{X} = \mathcal{D}\mathcal{V}^\mathsf{T}$. On a $\mathrm{var}(Y) = \mathcal{E}^\mathsf{T}\mathrm{var}(\dot{X})\mathcal{E} = \mathcal{E}^\mathsf{T} S_{\mathrm{X}}\mathcal{E} = \mathcal{E}^\mathsf{T}\frac{1}{v_m}\mathcal{E}\Lambda\mathcal{E}^\mathsf{T}\mathcal{E} = \frac{1}{v_m}\Lambda = \frac{1}{v_m}\mathcal{D}^2$.
Posons alors $Z = \sqrt{v_m}\mathcal{D}^{-1}Y = \sqrt{v_m}\mathcal{V}^\mathsf{T}$. Alors $\mathrm{var}(Z) = \mathcal{I}$.

On applique alors un algorithme d'ACIs sur Z, ce qui consiste à chercher $\hat{\mathcal{W}}_{\mathrm{Z}}$ (et $\hat{\mathcal{A}}_{\mathrm{Z}}$) telle que les sources estimées $\hat{S} = \hat{\mathcal{W}}_{\mathrm{Z}}Z$ soient les plus indépendantes possible.

Notons $\hat{\mathcal{W}}_{\dot{\mathrm{X}}}$ (resp. $\hat{\mathcal{A}}_{\dot{\mathrm{X}}}$) la matrice de séparation (resp. de mélange) que l'on aurait obtenue en appliquant un algorithme d'ACIs sur $\dot{X}$:

$$\dot{X} = \hat{\mathcal{A}}_{\dot{\mathrm{X}}}\hat{S} \tag{7.3}$$

et

$$\hat{S} = \hat{\mathcal{W}}_{\dot{\mathrm{X}}}\dot{X}. \tag{7.4}$$

On a $\dot{X} = \mathcal{U}\mathcal{D}\mathcal{V}^\intercal = \hat{\mathcal{A}}_{\dot{\mathrm{X}}}\hat{S} = \hat{\mathcal{A}}_{\dot{\mathrm{X}}}\hat{\mathcal{W}}_{\mathrm{Z}}Z = \hat{\mathcal{A}}_{\dot{\mathrm{X}}}\hat{\mathcal{W}}_{\mathrm{Z}}\sqrt{v_m}\mathcal{V}^\intercal$. De ceci, on tire

$$\hat{\mathcal{A}}_{\dot{\mathrm{X}}} = \mathcal{U}\mathcal{D}\mathcal{V}^\intercal\mathcal{V}\frac{1}{\sqrt{v_m}}\hat{\mathcal{W}}_{\mathrm{Z}}^{-1} = \frac{1}{\sqrt{v_m}}\mathcal{U}\mathcal{D}\hat{\mathcal{W}}_{\mathrm{Z}}^{-1} \tag{7.5}$$

et

$$\hat{\mathcal{W}}_{\dot{\mathrm{X}}} = \hat{\mathcal{A}}_{\dot{\mathrm{X}}}^{-1} = \sqrt{v_m}\hat{\mathcal{W}}_{\mathrm{Z}}\mathcal{D}^{-1}\mathcal{U}^\intercal. \tag{7.6}$$

On peut écrire

$$\dot{X} = \sum_{j=1}^{t_m} \hat{\mathcal{A}}_{\dot{\mathrm{X}}_{.j}} \otimes \hat{S}_{j.} \tag{7.7}$$

La matrice des données $\dot{X}$ s'écrit comme une somme de termes qui sont aussi des matrices. Chacun des termes est un produit extérieur faisant intervenir une composante ACIs ($\hat{S}_{j.}$) et son décours temporel associé ($\hat{\mathcal{A}}_{\dot{\mathrm{X}}_{.j}}$). Cette décomposition est à rapprocher de la décomposition DVS vue en ACP.

Pour de nombreux jeux de données, il commence à devenir peu praticable (pour des questions d'insuffisance de mémoire virtuelle sur les PC standards) de trouver une matrice de séparation $\hat{\mathcal{W}}_{\dot{\mathrm{X}}}$ de taille $t_m \times t_m$ car le nombre t_m de lignes dans $\dot{X}$ est trop grand. Dans de tels cas, l'ACPs peut être utilisée pour réduire la taille de $\mathcal{W}_{\dot{\mathrm{X}}}$.

On peut en effet utiliser la DVS tronquée et appliquer l'ACIs sur la matrice $Z_{red} = \sqrt{v_m}\mathcal{V}_{red}^\intercal$ de taille $r \times v_m$ (les premières lignes correspondant aux plus grandes valeurs singulières de la matrice $\mathcal{D}$) pour estimer une matrice de sources $\hat{S}_{red}$ de taille $r \times v_m$ où les matrices de mélange et de séparation associées sont $\hat{\mathcal{W}}_{\mathrm{Z}_{red}}$ et $\hat{\mathcal{A}}_{\mathrm{Z}_{red}}$ de taille $r \times r$. De ceci, on tire

$$\hat{\mathcal{A}}_{\dot{\mathrm{X}}_{red}} = \frac{1}{\sqrt{v_m}}\mathcal{U}_{red}\mathcal{D}_{red}\hat{\mathcal{W}}_{\mathrm{Z}_{red}}^{-1} = \hat{\mathcal{W}}_{\dot{\mathrm{X}}_{red}}^\intercal\left(\hat{\mathcal{W}}_{\dot{\mathrm{X}}_{red}}\hat{\mathcal{W}}_{\dot{\mathrm{X}}_{red}}^\intercal\right)^{-1}, \qquad \text{de taille } t_m \times r \tag{7.8}$$

et

$$\hat{\mathcal{W}}_{\dot{\mathrm{X}}_{red}} = \sqrt{v_m}\hat{\mathcal{W}}_{\mathrm{Z}_{red}}\mathcal{D}_{red}^{-1}\mathcal{U}_{red}^\intercal \quad , \qquad \text{de taille } r \times t_m. \tag{7.9}$$

Remarque 9. Notons qu'utiliser la DVS de cette manière nécessite l'hypothèse que les ICs ne sont pas distribuées parmi les plus petits vecteurs propres qui sont en général écartés. La validité de cette hypothèse n'est en aucune manière garantie.

Dans ce cas, on peut écrire (attention, ce n'est pas la somme des premiers termes de (7.7))

$$\dot{X}_{red} = \sum_{j=1}^{r} \hat{\mathcal{A}}_{\dot{\mathrm{X}}_{red,.j}} \otimes \hat{S}_{red,j.}. \tag{7.10}$$

Le dessin suivant illustre ce qui précède.

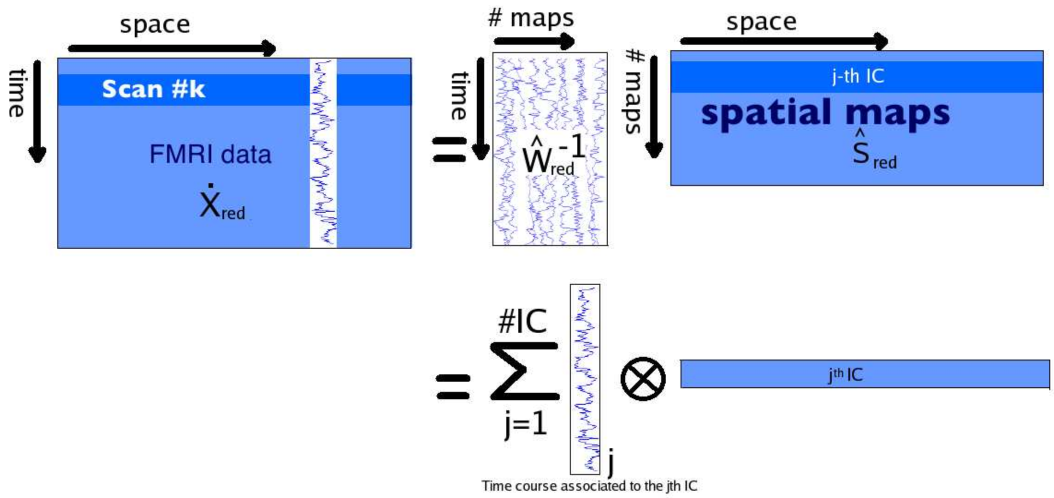


FIG. 11 – Illustration de la décomposition par l'ACIs d'un jeu de données en un certain nombre de cartes spatiales indépendantes et de leur décours tempoel associé.

7.6 Appliquer une ACIt après une ACPt

Dans toute cette sous-section, $\dot{X}$ désigne la matrice X de taille $t_m \times v_m$ qui a été centrée d'abord en lignes puis en colonnes. Cela correspond à centrer la matrice X^{T} d'abord en colonnes puis en lignes.

Ici, puisque dans l'analyse de signaux IRMF on a $t_m << v_m$, alors le nuage des t_m individus qui vit dans un espace à v_m dimensions est en fait situé dans un sous-espace de dimension au plus $t_m - 1$. Par conséquent, il y a au plus $t_m - 1$ valeurs propres non nulles de $\text{var}(\dot{X}^{\mathsf{T}})$.

L'idée est ici de commencer par effectuer une ACPt de $\dot{X}^{\mathsf{T}}$, pour utiliser une ACIt non pas sur $\dot{X}^{\mathsf{T}}$ mais sur la matrice des r premières composantes principales (ou plutôt une version normalisée de celle-ci).
Pour effectuer une ACPt de $\dot{X}^{\mathsf{T}}$, on se servira du Résultat 1 en page 44 des annexes qui permet d'obtenir les r premières composantes principales de l'ACPt à partir de celles de l'ACPs.
Notons $\mathcal{U}_r$ de taille $t_m \times r$ (resp. $\mathcal{D}_r^2$ de taille $r \times r$) la matrice des r premiers vecteurs propres (resp. premières valeurs propres) de $v_m \text{var}(\dot{X}) = \dot{X}\dot{X}^{\mathsf{T}}$. Posons alors $Z = \sqrt{t_m}\mathcal{D}_r^{-1}\mathcal{V}_r^{\mathsf{T}}\dot{\mathcal{X}}^{\mathsf{T}} = \sqrt{t_m}\mathcal{U}_r^{\mathsf{T}}$ qui est telle que $\text{var}(Z) = \mathcal{I}_r$, et où $\mathcal{V}_r$ est la matrice des r premiers vecteurs propres de $t_m \text{var}(\dot{X}^{\mathsf{T}}) = \dot{X}^{\mathsf{T}}\dot{X}$. Notons que l'on a la relation $\mathcal{V}_r = \dot{X}^{\mathsf{T}}\mathcal{U}_r\mathcal{D}_r^{-1}$.

La matrice $\sqrt{t_m}\mathcal{D}_r^{-1}\mathcal{V}_r^{\mathsf{T}}$ contient dans chacune de ses lignes les r vecteurs de la base du sous-espace dans laquelle sont projetées les données. Ces vecteurs expriment une combinaison linéaire des axes du repère d'origine. On appelera ces nouveaux axes des pseudo-voxels puisque ce sont des combinaisons linéaires des v_m voxels d'origine, les coefficients de la combinaison linéaire exprimant le poids de chacun des voxels d'origine dans le pseudo-voxel correspondant.

On applique alors un algorithme d'ACIt sur Z, ce qui consiste à chercher une matrice de séparation $\hat{\mathcal{W}}_{\mathrm{Z}}$ de taille $r \times r$ et une matrice de mélange $\hat{\mathcal{A}}_{\mathrm{Z}}$ de taille $r \times r$ telles que les (pseudo)-sources estimées $\hat{S}_{pseudo} = \hat{\mathcal{W}}_{\mathrm{Z}}Z$ soient les plus indépendantes possible. Ici, chacune des r lignes de $\hat{S}_{pseudo}$ est une composante indépendante consistant en un décours temporel de longueur t_m. Chacune des r colonnes associées de la matrice $\hat{\mathcal{A}}_{\mathrm{Z}}$ contient les coefficients de mélange de la composante temporelle dans les pseudo-voxels.

Les cartes spatiales associées sont dans les colonnes de la matrice $\hat{\mathcal{A}}_{\dot{\mathcal{X}}^\mathsf{T}} = \sqrt{t_m}\mathcal{V}_r\mathcal{D}_r^{-1}\hat{\mathcal{W}}_Z^\mathsf{T}$.

7.7 La DVS en relation avec l'ACIs et l'ACIt

Si l'on distribue les valeurs singulières entre $\mathcal{U}$ et $\mathcal{V}$ dans (5.14) en multipliant chaque colonne dans $\mathcal{U}$ et $\mathcal{V}$ par la racine carrée de la valeur singulière correspondante alors on a

$$\dot{X} = \mathcal{U}\mathcal{V}^\mathsf{T} \tag{7.11}$$

qui a une forme similaire à la décomposition ACI

$$\dot{X} = \hat{\mathcal{A}}\hat{S}. \tag{7.12}$$

La différence principale entre la DVS et l'ACI est la suivante. Chaque matrice produite par la DVS a des colonnes orthogonales. C'est-à-dire que la variation dans chaque colonne de $\mathcal{U}$ et $\mathcal{V}$ est non corrélée avec les variations de toute autre colonne de $\mathcal{U}$ et $\mathcal{V}$. A l'inverse, l'ACI produit deux matrices avec des propriétés assez différentes. Plutôt que d'être non corrélées, les lignes de $\hat{S}$ sont (le plus possible) indépendantes. La contrainte plus forte sur les lignes de $\hat{S}$ suggère que les colonnes de $\hat{\mathcal{A}}$ ne peuvent pas aussi être indépendantes, et l'ACI ne place donc aucune contrainte sur les relations entre les colonnes de $\hat{\mathcal{A}}$.

Contrastant avec la décomposition obtenue par la DVS, les composantes produites par l'ACI ne présentent pas d'ordre imposé. Pour classer les composantes indépendantes, on peut déterminer la norme des colonnes de la matrice de mélange $\hat{\mathcal{A}}$ ou bien aussi se baser sur la néguentropie.

8 La méthode de Poursuite de Projection (PP)

L'hypothèse explicite d'indépendance sur laquelle repose l'ACI est moins sévère que l'hypothèse impliquant la non gaussianité des signaux sources. On peut montrer [MG96] que, étant donné un ensemble de mélange de signaux non gaussiens indépendants, les sources peuvent être extraites en trouvant des composantes qui ont des fonctions de répartition appropriées, et que ces signaux sont indépendants. L'implication inverse n'est pas vraie, en général, si le nombre de signaux extraits est plus petit que le nombre de signaux indépendants dans l'ensemble des signaux mélange. C'est-à-dire, simplement trouver un sous-ensemble de signaux indépendants dans un ensemble de mélanges de signaux sources indépendants non gaussiens n'est pas, en général, équivalent à trouver les composantes sources. Cela provient du fait que des combinaisons linéaires d'ensembles disjoints de signaux sources sont indépendantes. Par exemple, si un sous ensemble de signaux indépendants sont combinés pour former un signal mélange x_1, et un sous ensemble disjoint d'autres signaux sont combinés pour former un mélange x_2, alors x_1 et x_2 sont mutuellement indépendants, même si ils sont tous les deux constitués de mélanges de signaux sources. Ainsi, l'indépendance statistique de signaux extraits est une condition nécessaire mais non suffisante, pour la séparation de sources.

Une caractéristique cruciale d'un mélange linéaire de signaux aléatoires (de variance finie) est que l'histogramme de ses valeurs est approximativement gaussien. Cela provient du théorème central limite. La plupart des mélanges d'un ensemble de signaux sont gaussiens. Comme les méthodes de séparation de sources utilisent un mélange en entrée, et produisent une somme pondérée en sortie, il s'ensuit que des matrices de séparation arbitraires $\mathcal{W}$ produisent aussi des signaux gaussiens. Cependant, si une matrice de séparation produisant un signal non gaussien à partir d'un ensemble de mélanges existe, alors un tel signal a peu de chances d'être un mélange de signaux.

Si l'on peut supposer que les signaux sources ont des densités non gaussiennes alors, tandis que la plupart des transformations (appliquées sur les mélanges) produisent des données avec des distributions gaussiennes, un petit nombre de transformations existent qui produisent des données avec une distribution non gaussienne.

Sous certaines conditions, les signaux non gaussiens extraits d'un mélange de signaux par une telle transformation sont en fait les signaux sources originaux. Ceci est la base des méthodes de poursuite de projection. Commençons par préciser ce que signifie le terme "non gaussien". Deux classes importantes de signaux ayant une distribution non gaussienne sont les distributions super-gaussiennes ou sous-gaussiennes. Ces classes sont caractérisées par leur kurtosis κ :

$$\kappa = \frac{\mathbf{E}[(\bar{\text{s}} - \text{s})^4]}{\mathbf{E}[(\bar{\text{s}} - \text{s})^2]} - 3 \tag{8.1}$$

où $\bar{\text{s}}$ est la valeur moyenne de s_t et la constante 3 assure que les signaux super-gaussiens ont un kurtosis positif, alors que les signaux sous-gaussiens ont un kurtosis négatif.
Un signal de distribution super-gaussienne possède la plupart de ses valeurs concentrées autour de 0, alors qu'un signal de loi sous-gaussienne a des ailes plus relevées.

Les méthodes PP utilisent les moments d'ordre supérieur (comme le kurtosis) pour estimer dans quelle mesure un signal est gaussien. Cependant, on peut utiliser une mesure plus générale empruntée à l'ACI. Le degré de non gaussianité d'un signal dépend des observations cruciales suivantes : si les valeurs scalaires d'un signal s sont transformées par la fonction de répartition F de ce signal alors la loi en résultant sera uniforme. Ceci est utile car cela permet de traduire l'écart à la gaussianité en écart à l'uniformité, ou de façon équivalente, l'entropie du signal transformé, $\text{Y} = F(\text{s})$ (une façon de voir l'entropie est comme une mesure de l'uniformité d'une distribution donnée).
La question de trouver la transformation linéaire capable de retrouver un signal source découle de la définition de notre mesure d'écart à la gaussianité. On a établit qu'une transformation linéaire $\mathcal{W}$ existe telle qu'un signal $\text{s} = \mathcal{W}\mathbf{x}$ peut être retrouvé à partir d'un ensemble de v_m signaux mélangés $\mathbf{x}$, et que cette transformation produit un signal $\mathbf{s} = \mathcal{W}\mathbf{x}$ tel que $\mathbf{s} = F(\mathbf{x})$ a une entropie maximale. En inversant la logique de cet argument, il s'ensuit que $\mathbf{s}$ peut être retrouvé à partir de $\mathbf{x}$ en trouvant un $\mathcal{W}$ qui maximise l'entropie $H(\mathbf{Y})$ de $\mathbf{s} = F(\mathcal{W}\mathbf{x})$.
On peut montrer [MG96] que si un certain nombre de signaux sont extraits d'un mélange $\mathbf{x}$ (et ceux-ci sont les composantes les plus non gaussiennes du mélange) alors on est garanti qu'ils seront mutuellement indépendants. Ainsi, même si une mesure de dépendance n'est pas explicitement maximisée pendant le processus d'optimisation de la méthode PP, extraire des signaux non gaussiens produit des signaux qui sont mutuellement indépendants. D'un autre côté, l'ACI maximise de façon explicite l'indépendance mutuelle entre les signaux extraits.

9 Conclusion et perspectives

Nous avons démontré dans ce travail comment des méthodes non paramétriques multivariées telles que l'Analyse en Composantes Principales, l'Analyse en Composantes Indépendantes et la Poursuite de Projection pouvaient être utilisées pour l'analyse des images d'IRMF. L'Analyse en Composantes Indépendantes temporelle, considérée comme impraticable dans la littérature pour l'étude des données d'IRMF, a pu être appliquée ici en projetant préalablement les données dans un sous-espace contenant la plus grande partie de l'information présente dans les données. Une relation liant le domaine spatial au domaine temporel dans la Décomposition aux Valeurs Singulières a rendu cette projection possible malgré la grande taille des matrices en jeu. Nous espérons que dans le futur cette voie pourra être empruntée pour détecter des signaux temporels intéressants qui ne pouvaient être obtenus au moyen de l'ACI spatiale.

Ce travail préliminaire laisse toutefois de nombreuses questions en suspens ouvrant d'intéressantes perspectives de recherche.

Les points suivants sont assez fondamentaux et nécessiteront de mener des recherches approfondies :

- Nous avons remarqué qu'un filtrage préalable des données (soustraction de tendance linéaire par exemple) pouvait s'avérer utile car il permettait de supprimer des composantes temporelles inintéressantes. De nombreuses autres techniques de filtrage existent, comme les méthodes d'ondelettes par exemple, et il sera bon de voir si elles peuvent apporter une amélioration à la détection des composantes d'intérêt.
- La sélection des composantes temporelles intéressantes se fait actuellement par une corrélation entre le signal du paradigme (convolué avec la HRF) et le décours temporel en cours d'investigation. Il semble intéressant de voir ce que peuvent donner des mesures de dépendance plus sophistiquées que la corrélation, qui permettraient peut-être de limiter les effets de l'*a priori* imposé par le modèle de HRF.
- Les composantes trouvées par l'ACI ne sont pas ordonnées. Il serait intéressant de trouver un critère conjuguant néguentropie et variance pour parvenir à les classer.
- Nous avons négligé, dans l'application de l'ACI, la répartition spatiale des voxels. Une option pour prendre en compte cette information pourrait consister à utiliser une décomposition aux valeurs singulières d'ordre supérieur. Les travaux de [Bih01] et [L.97] pourront s'avérer utile.
- Le signal mesuré en IRMF est toujours positif. Par conséquent, nous pourrions essayer d'appliquer une méthode d'ACI positive comme celle introduite par [Plu02].
- L'article de [AAG03] ouvre la possibilité de détecter les clusters de points sur le graphique 5 de la page 19, ce qui permettrait ainsi de retrouver des composantes spatiales intéressantes. Un test-t multivarié permettrait ensuite d'obtenir des p-valeurs associées.
- Nous avons utilisé ces méthodes sur un seul sujet. Il s'agira de voir si certains problèmes apparaissent lors d'une étude de groupe et si nous sommes capables de proposer des méthodes offrant une certaine reproductibilité. Celles-ci devraient pouvoir permettre de dégager ce qui a spécifiquement trait à l'individu et ce qui au contraire se retrouve chez tous les sujets.
- Les méthodes d'ACI nécessitent d'optimiser une certaine mesure de dépendance. Il sera intéressant de voir comment la mesure de dépendance multivariée proposée par l'auteur de ce mémoire dans [BLdM05] pourra être utilisée.

On peut aussi signaler certains points techniques que nous avons laissés en suspens faute de temps :

- Après avoir trouvé une composante spatiale indépendante, il s'agit de sélectionner les voxels de cette carte qui peuvent être considérés comme actifs. Pour cela, nous avons utilisé un simple test de Student reposant sur l'hypothèse de gaussianité des observations. Ceci est en contradiction avec l'hypothèse de non gaussianité des sources nécessaire en ACI ou en PP. Les travaux de [BS04] peuvent être mentionnés comme une première solution à ce problème.
- Le choix du nombre de composantes à calculer est une étape cruciale de l'algorithme. Si on le prend trop petit, on risque de manquer des composantes intéressantes. Si on le prend trop grand, la puissance de

détection s'en trouve amoindrie et le temps de calcul peut devenir prohibitif.

– Un des gros problèmes lors de l'utilisation de ce type de méthodes statistiques réside dans la grande taille des matrices de données à traiter. Une option pourrait consister à travailler sur les données projetées sur une carte dépliée du cortex. Ainsi, toutes les données beaucoup moins pertinentes de la matière blanche seraient éliminées, réduisant d'autant la taille du jeu de données à traiter. Cette approche peut être utile pour les cas où l'on dispose d'une topie corticale de la fonction cognitive étudiée. Une autre option consisterait à diminuer la résolution des mesures dans les zones où le signal est homogène. Une résolution multi-échelle donnerait probablement de bons résultats.

Bibliographie

[AAG03] Umberto Amato, Anestis Antoniadis, and Gérard Grégoire. Independent component discriminant analysis. *Int. Math. J.*, 3(7) :735–753, 2003.

[AGA99] Anders H. Andersen, Don M. Gash, and Malcolm J. Avison. Principal component analysis of the dynamic response measured by fMRI : A generalized linear systems framework. *Magnetic Resonance Imaging*, 17(6) :795–815, July 1999.

[AR83] John R. Anderson and Richard D. Rosen. The latitude-height structure of 40-50 day variations in atmospheric angular momentum. *Journal of the Atmospheric Sciences*, 40 :1584–1591, June 1983.

[AS07] K. Paliwal A. Sharma. Fast principal component analysis using fixed-point algorithm. *Pattern Recognition Letters*, 28 :1151–1155, 2007.

[Bih01] Nicolas Le Bihan. *Traitement algébrique des signaux vectoriels. Application en séparation d'ondes sismiques.* PhD thesis, INPG, 2001.

[BLdM05] M. Bilodeau and P. Lafaye de Micheaux. A multivariate empirical characteristic function test of independence with normal marginals. *J. Multivariate Anal.*, 95(2) :345–369, 2005.

[BS95] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Technical report, Institute for Neural Computation, UCSD, San Diego, CA 92093-0523, 1995.

[BS04] Christian F Beckmann and Stephen M Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans Med Imaging*, 23(2) :137–152, Feb 2004. Evaluation Studies.

[BS05] C F Beckmann and S M Smith. Tensorial extensions of independent component analysis for multisubject FMRI analysis. *Neuroimage*, 25(1) :294–311, Mar 2005.

[BYS01] G D Brown, S Yamada, and T J Sejnowski. Independent component analysis at the neural cocktail party. *Trends Neurosci*, 24(1) :54–63, Jan 2001.

[BZ04a] Andreas Bartels and Semir Zeki. Functional brain mapping during free viewing of natural scenes. *Hum Brain Mapp*, 21(2) :75–85, Feb 2004.

[BZ04b] Andreas Bartels and Semir Zeki. The chronoarchitecture of the human brain–natural viewing conditions reveal a time-based anatomy of the brain. *Neuroimage*, 22(1) :419–433, May 2004.

[BZ05] Andreas Bartels and Semir Zeki. Brain dynamics during natural viewing conditions–a new guide for mapping connectivity in vivo. *Neuroimage*, 24(2) :339–349, Jan 2005.

[CA06] Vince D Calhoun and Tulay Adali. Unmixing fMRI with independent component analysis. *IEEE Eng Med Biol Mag*, 25(2) :79–90, Mar 2006.

[CAM$^+$01] V D Calhoun, T Adali, V B McGinty, J J Pekar, T D Watson, and G D Pearlson. fMRI activation in a visual-perception task : network of areas detected using the general linear model and independent components analysis. *Neuroimage*, 14(5) :1080–1088, Nov 2001.

[CAP$^+$02] V D Calhoun, T Adali, G D Pearlson, P C M van Zijl, and J J Pekar. Independent component analysis of fMRI data in the complex domain. *Magn Reson Med*, 48(1) :180–192, Jul 2002.

[CAP04] Vince D Calhoun, Tulay Adali, and James J Pekar. A method for comparing group fMRI data using independent component analysis : application to visual, motor and visuomotor tasks. *Magnetic Resonance Imaging*, 22(9) :1181–1191, Nov 2004. Comparative Study.

[CAPP01a] V D Calhoun, T Adali, G D Pearlson, and J J Pekar. A method for making group inferences from functional MRI data using independent component analysis. *Human Brain Mapping*, 14(3) :140–151, Nov 2001.

[CAPP01b] V D Calhoun, T Adali, G D Pearlson, and J J Pekar. Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms. *Hum Brain Mapp*, 13(1) :43–53, May 2001.

[CAPP03] V D Calhoun, T Adali, J J Pekar, and G D Pearlson. Latency (in)sensitive ICA. Group independent component analysis of fMRI data in the temporal frequency domain. *Neuroimage*, 20(3) :1661–1669, Nov 2003. Clinical Trial.

[CAS+05] V D Calhoun, T Adali, M C Stevens, K A Kiehl, and J J Pekar. Semi-blind ICA of fMRI : A method for utilizing hypothesis-derived time courses in a spatial ICA analysis. *Neuroimage*, 25(2) :527–538, Apr 2005.

[Com94] Pierre Comon. Independent component analysis, a new concept ? *Signal Processing*, 36 :287–314, 1994.

[Cra65] J. M. Craddock. A meterological application of principal component analysis. *The Statistician*, 15(2) :143–156, 1965.

[DR03] Pierre Duchesne and Roch Roy. Robust tests for independence of two time series. *Statist. Sinica*, 13(3) :827–852, 2003.

[ESH+05] Fabrizio Esposito, Tommaso Scarabino, Aapo Hyvarinen, Johan Himberg, Elia Formisano, Silvia Comani, Gioacchino Tedeschi, Rainer Goebel, Erich Seifritz, and Francesco Di Salle. Independent component analysis of fMRI group studies by self-organizing clustering. *Neuroimage*, 25(1) :193–205, Mar 2005.

[FBLK02] Ola Friman, Magnus Borga, Peter Lundberg, and Hans Knutsson. Exploratory fMRI analysis by autocorrelation maximization. *Neuroimage*, 16(2) :454–464, Jun 2002. Comparative Study.

[FEDSG04] Elia Formisano, Fabrizio Esposito, Francesco Di Salle, and Rainer Goebel. Cortex-based independent component analysis of fMRI time series. *Magn Reson Imaging*, 22(10) :1493–1504, Dec 2004. Comparative Study.

[Fri98] Karl J. Friston. Modes or models : a critique on independent component analysis for fmri. *Trends in Cognitive Sciences*, 2 :373–375, 1998.

[Fri03a] Ola Friman. *Adaptive Analysis of Functional MRI Data*. PhD thesis, Linköpings universitet, 2003.

[Fri03b] K. Friston. Statistical parametric mapping, the wellcome department of cognitive neurology, london ; http ://www.fil.ion.ucl.ac.uk/spm, 2003.

[GMR04] C. W. Granger, E. Maasoumi, and J. Racine. A dependence metric for possibly nonlinear processes. *J. Time Ser. Anal.*, 25(5) :649–669, 2004.

[Gro97] D. Groutage. A fast algorithm for computing minimum cross-entropy positive time-frequency distributions. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 45(8) :1954–1970, August 1997.

[HHE04] Johan Himberg, Aapo Hyvarinen, and Fabrizio Esposito. Validating the independent components of neuroimaging time series via clustering and visualization. *Neuroimage*, 22(3) :1214–1222, Jul 2004.

[HHS06] H Hinrichs, H J Heinze, and M A Schoenfeld. Causal visual interactions as revealed by an information theoretic measure and fMRI. *Neuroimage*, 31(3) :1051–1060, Jul 2006.

[HKO01] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley-Interscience, May 2001.

[HMTM02] Olivier Houdé, Bernard Mazoyer, and Nathalie Tzourio-Mazoyer. *Cerveau et Psychologie*. Presses Universitaires de France, 2002.

[HO00] A Hyvarinen and E Oja. Independent component analysis : algorithms and applications. *Neural Netw*, 13(4-5) :411–430, May 2000.

[HYL+05] Dewen Hu, Lirong Yan, Yadong Liu, Zongtan Zhou, Karl J Friston, Changlian Tan, and Daxing Wu. Unified SPM-ICA for fMRI analysis. *Neuroimage*, 25(3) :746–755, Apr 2005.

[JVSH00] NR Porter James V Stone, John Porrill and NM Hunkin. Spatiotemporal ica of fmri data. Technical report, Psychologie Department, Sheffield University, Sheffield, England, S302TP, 2000.

[KF94] R. Turner K. Friston, P. Jezzard. Analysis of functional mri time-series. *Human Brain Mapping*, 1 :153–171, 1994.

[KFB99] Dave Chawla Karl Friston, Jacquie Phillips and Christian Buchel. Revealing interactions among brain systems with nonlinear pca. *Human Brain Mapping*, 8 :92–97, 1999.

[KFB00] Dave Chawla Karl Friston, Jacquie Phillips and Christian Buchel. Nonlinear pca : characterizing interactions between modes of brain activity. *Philosophical Transactions of the Royal Society of London B*, 355 :135–146, 2000.

[KL05] Eunhee Kim and Sangyeol Lee. A test for independence of two stationary infinite order autoregressive processes. *Ann. Inst. Statist. Math.*, 57(1) :105–127, 2005.

[L.97] De Lathauwer L. *Signal Processing based on Multilinear Algebra*. PhD thesis, Faculty of Engineering, K.U.Leuven (Leuven, Belgium), 1997.

[LF99] Shang-Hong Lai and Ming Fang. A novel local pca-based method for detecting activation signals in fmri. *Magnetic Resonance Imaging*, 17(6) :827–836, 1999.

[LSA+99] N Lange, S C Strother, J R Anderson, F A Nielsen, A P Holmes, T Kolenda, R Savoy, and L K Hansen. Plurality and resemblance in fMRI data analysis. *Neuroimage*, 10(3 Pt 1) :282–303, Sep 1999.

[McK00] M J McKeown. Detection of consistently task-related activations in fMRI data with hybrid independent component analysis. *Neuroimage*, 11(1) :24–35, Jan 2000.

[MG96] C. Fyfe M. Girolami. Negentropy and kurtosis as projection pursuit indices provide generalised ica algorithms. *NIPS96 Blind Signal Separation Workshop*, 1996.

[MHH07] Sanna Malinen, Yevhen Hlushchuk, and Riitta Hari. Towards natural stimulation in fMRI–issues of data analysis. *Neuroimage*, 35(1) :131–139, Mar 2007.

[MHS03] Martin J McKeown, Lars Kai Hansen, and Terrence J Sejnowski. Independent component analysis of functional MRI : what is signal and what is noise ? *Curr Opin Neurobiol*, 13(5) :620–629, Oct 2003.

[MMB+98] M J McKeown, S Makeig, G G Brown, T P Jung, S S Kindermann, A J Bell, and T J Sejnowski. Analysis of fMRI data by blind separation into independent spatial components. *Hum Brain Mapp*, 6(3) :160–188, 1998. Clinical Trial.

[MS98] M J McKeown and T J Sejnowski. Independent component analysis of fMRI data : examining the assumptions. *Hum Brain Mapp*, 6(5-6) :368–372, 1998. Clinical Trial.

[MWAH06] Martin J McKeown, Z Jane Wang, Rafeef Abugharbieh, and Todd C Handy. Increasing the effect size in event-related fMRI studies. *IEEE Eng Med Biol Mag*, 25(2) :91–101, Mar 2006.

[NMB+04] Toshiharu Nakai, Shigeru Muraki, Epifanio Bagarinao, Yukio Miki, Yasuo Takehara, Kayako Matsuo, Chikako Kato, Harumi Sakahara, and Haruo Isoda. Application of independent component analysis to magnetic resonance imaging for enhancing the contrast of gray and white matter. *Neuroimage*, 21(1) :251–260, Jan 2004.

[PBA+07] Vincent Perlbarg, Pierre Bellec, Jean-Luc Anton, Melanie Pelegrini-Issac, Julien Doyon, and Habib Benali. CORSICA : correction of structured noise in fMRI by automatic identification of ICA components. *Magn Reson Imaging*, 25(1) :35–46, Jan 2007. Evaluation Studies.

[Pek06] James J Pekar. A brief introduction to functional MRI. *IEEE Eng Med Biol Mag*, 25(2) :24–26, Mar 2006.

[Plu02] M. Plumbley. Conditions for nonnegative independent component analysis. *IEEE Signal Processing Letters*, 9(6) :177–180, June 2002.

[PMI+96] S M Pincus, T Mulligan, A Iranmanesh, S Gheorghiu, M Godschalk, and J D Veldhuis. Older males secrete luteinizing hormone and testosterone more irregularly, and jointly more asynchronously, than younger males. *Proc Natl Acad Sci U S A*, 93(24) :14100–14105, Nov 1996. Comparative Study.

[PS96] S Pincus and B H Singer. Randomness and degrees of irregularity. *Proc Natl Acad Sci U S A*, 93(5) :2083–2088, Mar 1996.

[PT95] D. Prichard and J. Theiler. Generalized redundancies for time series analysis. *Physica D*, 84 :476–493, 1995.

[R D06] R Development Core Team. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[RM00] J S Richman and J R Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *Am J Physiol Heart Circ Physiol*, 278(6) :2039–2049, Jun 2000.

[RM04] Marc M. Van Hulle R. Mutihac. Comparison of principal component analysis and independent component analysis for blind source separation. *Romanian Reports in Physics*, 56(1) :20–32, 2004.

[RTZ+06] Jagath C Rajapakse, Choong Leong Tan, Xuebin Zheng, Susanta Mukhopadhyay, and Kanyan Yang. Exploratory analysis of brain connectivity with ICA. *IEEE Eng Med Biol Mag*, 25(2) :102–111, Mar 2006.

[SKB02] Markus Svensen, Frithjof Kruggel, and Habib Benali. ICA of fMRI group study data. *Neuroimage*, 16(3 Pt 1) :551–563, Jul 2002.

[SLML06] K. Sjöstrand, T. E. Lund, K. H. Madsen, and R. Larsen. Sparse PCA, a new method for unsupervised analyses of fmri data. In *Proc. International Society of Magnetic Resonance In Medicine - ISMRM 2006, Seattle, Washington, USA*, pages 1–1, Berkeley, CA, USA, may 2006. ISMRM.

[SP98] James V Stone and John Porrill. Independent component analysis and projection pursuit : A tutorial introduction. Technical report, Psychologie Department, Sheffield University, Sheffield, England, 1998.

[Sto02] James V Stone. Independent component analysis : an introduction. *Trends Cogn Sci*, 6(2) :59–64, Feb 2002.

[TF03] Bertrand Thirion and Olivier Faugeras. Dynamical components analysis of fmri data through kernel pca. *NeuroImage*, 20 :34–49, 2003.

[THM02] Christopher G Thomas, Richard A Harshman, and Ravi S Menon. Noise reduction in BOLD-based fMRI using component analysis. *Neuroimage*, 17(3) :1521–1537, Nov 2002. Evaluation Studies.

[tMia99] Spatial temporal Modelling and its applications, editors. *Spatial, Temporal, and Spatiotemporal Independent Component Analysis of fMRI Data*, 1999.

[US06] M.O. Ulfarsson and V. Solo. Smooth principal component analysis with application to functional magnetic resonance imaging. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.*, 2 :993–996, 2006.

[VCP01] T. Adali V.D. Calhoun and G.D. Pearlson. ”independent components analysis applied to fmri data : A natural model and order selection”. In *Proc. NSIP, Baltimore MD*, pages 1–5, 2001.

[VDVBU06] Dimitri Van De Ville, Thierry Blu, and Michael Unser. Surfing the brain. *IEEE Eng Med Biol Mag*, 25(2) :65–78, Mar 2006.

[VGS05] Roberto Viviani, Georg Gron, and Manfred Spitzer. Functional principal component analysis of fMRI data. *Hum Brain Mapping*, 24(2) :109–129, Feb 2005.

[War02] Jan Warnking. *Délinéation des aires visuelles rétinotopiques chez l'homme par IRM fonctionnelle*. PhD thesis, Université Joseph Fourier, Grenoble, 2002.

[WRW+05] Ze Wang, Hengyi Rao, Jiongjiong Wang, John Detre, and Anna Childress. CRLS-PCA Based Independent Component Analysis for fMRI Study. *Conf Proc IEEE Eng Med Biol Soc*, 6 :5904–5907, 2005.

[ZKHB03] Gerhard Zuendorf, Nacer Kerrouche, Karl Herholz, and Jean-Claude Baron. Efficient principal component analysis for multivariate 3D voxel-based mapping of brain functional imaging data sets as applied to FDG-PET and normal aging. *Hum Brain Mapp*, 18(1) :13–21, Jan 2003.

Annexes

A L'imagerie cérébrale

Cette section fournit un bref survol de l'Imagerie par Résonance Magnétique (IRM) et de l'Imagerie par Résonance Magnétique Fonctionnelle (IRMF). L'IRM a été inventée au début des années 1970 par Paul Lauterbur, de l'Université d'Urbana dans l'Illinois (États-Unis). L'IRMF a été inventée dans les années 1990 par l'équipe de Bruce Rosen au Massachussetts General Hospital à Boston (États-Unis). Ces techniques permettent d'obtenir des images numériques en deux ou trois dimensions d'une précision inférieure au millimètre, du cortex, de la substance blanche, du liquide céphalo-rachidien et des noyaux gris centraux. Ce type d'acquisition permet d'effectuer une analyse neuroanatomique individuelle de très haute précision. Les bases physiques sur lesquelles reposent l'IRM et l'IRMF viennent du phénomène de Résonance Magnétique Nucléaire (RMN). La figure 12 montre la grande précision spatiale des images IRM.

FIG. 12 – IRM encéphalique (coupe sagittale passant par la ligne médiane)

Le lecteur intéressé pourra consulter avec profit les ouvrages suivants [HMTM02] et [Fri03a] desquels nous avons tiré la plupart des informations fournies dans cette section.

A.1 Résonance magnétique nucléaire

Chaque atome de matière est constitué de particules qui sont caractérisées par une propriété quantique intrinsèque appelée spin qui est caractéristique de la nature de la particule, au même titre que sa masse et sa charge électrique. Elle permet de caractériser le comportement de la particule sous l'effet de la symétrie de rotation de l'espace. Les protons, les neutrons et les électrons ont une valeur de spin égale à $1/2$. Le spin de particules composées, comme le noyau atomique ou l'atome, est constitué des spins des particules qui les composent auxquels s'ajoute le moment angulaire des particules élémentaires l'une par rapport à l'autre.
Dans la méthode par Résonance Magnétique Nucléaire, on utilise le spin des noyaux des atomes. Certains noyaux possèdent un spin nul (ceux dont le nombre de protons et de neutrons sont tous les deux pairs) mais d'autres ont un spin nucléaire différent de zéro, ce qui implique que l'on peut leur associer un moment magnétique de spin qui se comporte comme une sorte de petit aimant. Les atomes de carbone 12 et d'oxygène 16 sont très répandus dans la nature mais leur spin nucléaire est nul. En revanche l'hydrogène n'a qu'un proton

et son moment magnétique nucléaire est ainsi non nul : il vaut 1/2. Puisque les molécules d'eau sont constituées d'atomes d'hydrogène, il y a une abondance d'atomes d'hydrogène dans le corps humain pouvant être utilisés pour produire des images tomographiques. La résonance magnétique de l'hydrogène (du proton) est donc la plus utilisée. De plus, un tel atome interagit avec un champ magnétique extérieur qui lui est appliqué.

La physique quantique nous apprend qu'un moment magnétique de spin 1/2 placé dans un champ magnétique extérieur d'intensité B_0 peut avoir deux niveaux d'énergie possibles.

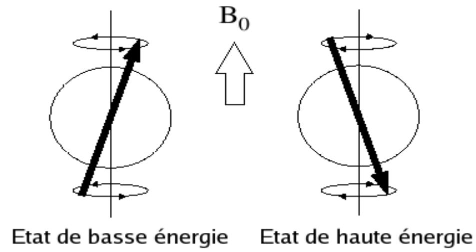


FIG. 13 – Les deux états d'énergie d'un moment magnétique de spin

La RMN consiste à modifier le moment magnétique nucléaire, autrement dit à faire passer le noyau d'un niveau d'énergie à un autre, (ce qui revient à "retourner" le spin) par absorption d'un photon : lorsque l'énergie du photon (et partant la fréquence de l'onde électromagnétique) permet cette transition il y a résonance. Pour les champs usuels utilisés en imagerie médicale (de l'ordre du tesla) la résonance du proton a lieu dans le domaine des ondes radio (100 MHz environ) : 42 MHz dans un champ de 1,0 T et 63 MHz dans un champ de 1,5 T.

La relation mathématique existant entre le champ magnétique imposé de norme B_0 et la fréquence de résonance (retournement de spin) ν_0 est très simple :

$$\nu_0 = \gamma \frac{B_0}{2\pi} \tag{A.1}$$

où γ est le rapport gyromagnétique caractéristique de chaque noyau étudié. Pour le noyau d'hydrogène, on a $\gamma = 42,58MHz/T$.

Lorsque l'on place la tête d'un sujet dans un champ magnétique $\boldsymbol{B}_0$, les moments magnétiques des noyaux d'hydrogène de celle-ci s'orientent selon l'axe de ce champ, soit dans le même sens, soit en sens inverse. Une faible majorité de spins étant orientés dans le sens du champ $\boldsymbol{B}_0$, il apparaît donc pour l'ensemble du cerveau un moment magnétique résultant non nul, noté $\boldsymbol{M}_0$, aligné avec $\boldsymbol{B}_0$. Tout se passe comme si, en plongeant le cerveau du sujet dans un champ magnétique, on l'avait aimanté.
Par ailleurs, l'application du champ magnétique $\boldsymbol{B}_0$ donnera aux noyaux d'hydrogène un moment de torsion qui forcera le moment magnétique à tourner (précession) autour de la direction du champ (voir Figure 13). La fréquence de cette rotation, appelée fréquence de Larmor, vaut $f_0 = 2\pi\nu_0$.

L'objectif d'une expérience d'IRM est de mesurer l'aimantation $\boldsymbol{M}_0$ en chaque point du cerveau. Comme $\boldsymbol{M}_0$ est aligné avec $\boldsymbol{B}_0$ mais que sa valeur est extrêmement faible, il faut l'écarter de l'axe de $\boldsymbol{B}_0$ pour pouvoir le mesurer. Pour ce faire, on utilise un autre champ magnétique, noté $\boldsymbol{B}_1$, perpendiculaire au champ $\boldsymbol{B}_0$ qui va exercer un moment de torsion sur $\boldsymbol{M}_0$. Pour que l'action de $\boldsymbol{B}_1$ sur $\boldsymbol{M}_0$ soit efficace, il faut que celui-ci tourne exactement à la fréquence propre de rotation des spins (fréquence de Larmor) : c'est la situation dite de "résonance". Il n'est pas nécessaire que l'intensité du champ $\boldsymbol{B}_1$ soit élevée, c'est le synchronisme entre $\boldsymbol{B}_1$ et $\boldsymbol{M}_0$ qui compte.

Compte tenu de sa fréquence, le champ magnétique $\boldsymbol{B}_1$ est une onde radio : il est créé par une antenne de radio-fréquence (RF) placée autour de la tête du sujet et maintenu pendant un temps très court (on parle d'impulsion) : la durée de l'impulsion détermine l'angle, dit de bascule, dont $\boldsymbol{M}_0$ s'écarte de l'axe $\boldsymbol{B}_0$ (sa valeur typique est de 90 degrés, ce qui correspond à une bascule de $\boldsymbol{M}_0$ dans le plan perpendiculaire à $\boldsymbol{B}_0$).

La transition du spin vers son retour à l'équilibre (la relaxation) entraîne l'émission d'une onde électromagnétique qui peut être détectée par un capteur.

En imagerie par résonance magnétique (IRM), on mesure ainsi la réponse de chaque voxel (volume de base de l'échantillon examiné), déterminant la densité en protons et reconstituant une image.

A.2 Le signal BOLD

Les cellules neuronales consomment de l'oxygène pour travailler. Cet oxygène est acheminé depuis les poumons vers les neurones en passant dans le sang où il est transporté par des molécules d'hémoglobine. La combinaison d'une molécule d'oxygène avec une molécule d'hémoglobine s'appelle de l'oxy-hémoglobine. Lorsque la molécule d'oxy-hémoglobine perd son oxygène on parle de déoxy-hémoglobine. Quand l'activité neuronale augmente, cela crée une demande en oxygène accrue qui se traduit par un afflux sanguin localisé au niveau de la population des neurones actifs. Ce mécanisme n'est pas totalement compris mais l'on sait cependant qu'un excès d'oxygène est fourni aux neurones atifs, ce qui mène à une augmentation locale de la concentration en oxy-hémoglobine. La séquence temporelle exacte de ces événements est la suivante. Au temps précoce (1 à 2 s), on observe une augmentation de la consommation d'oxygène, en rapport avec l'activité neuronale, qui génère une augmentation transitoire de la concentration en déoxy-hémoglobine. Puis le débit augmente de façon brutale, ce qui entraîne un afflux de globules rouges oxygénés et a pour effet d'abaisser la concentration relative en déoxy-hémoglobine : on appelle cela l'effet BOLD (*Blood Oxygen Level Dependent*). À l'arrêt de l'activation, le signal décroit en deçà de sa valeur de base car il persisterait temporairement une vasodilatation et une surconsommation d'oxygène sans augmentation de débit.

C'est cette différence de concentration en oxygène entre un niveau de base et un niveau actif qui peut être mesurée par un scanner à résonnance maégnétique grâce aux propriétés magnétiques de la molécule de déoxy-hémoglobine. Le signal BOLD est par conséquent un indicateur indirect de l'activité cérébrale et c'est lui qui constiue les images d'IRMF étudiées dans ce mémoire.

A.3 Détecter l'activité cérébrale

Pour effectuer une expérience d'IRMF, il est nécessaire d'utiliser un scanner à Résonance Magnétique capable d'acquérir des images d'Echo Planar Imaging (EPI). Un patient ou un volontaire est placé dans le scanner et des images EPI couvrant le volumes cérébral sont ensuite récoltées de façon continue durant une période de 5 à 10 minutes. La taille de chacune des images EPI est en général de 64×64 ou 128×128 et un empilement de 10 à 40 coupes sont collectées. Autour de 100 à 200 images volumiques de ce types sont successivement collectées pendant l'expérience avec une période d'échantillonnage comprise entre 1 et 5 secondes. Pendant que les images sont acquises, il est ou bien demandé au sujet d'effectuer une tâche cognitive ou alors il lui est présenté certains stimuli. Par exemple, un stimulus visuel peut être présenté sur un écran pendant 30 secondes, puis un écran blanc lui succède pour une autre période de 30 secondes. Ce bloc d'activité/repos est alors répété pendant toute la session. Grâce à l'effet BOLD, il y aura une une réponse BOLD dans les aires cérébrales qui sont activées par le stimulus visuel présenté. De telles aires seront probablement situées dans le cortex visuel. Ainsi, dans une image EPI collectée quand le stimulus visuel est présent, il devrait y avoir des valeurs d'intensité plus hautes dans les voxels couvrant des zones cérébrales actives que dans les mêmes voxels d'une image acquise pendant une période de repos. Comme l'image de la même coupe a été acquise plusieurs fois au cours du temps, on se retrouve avec une série temporelle de valeurs d'intensité dans chaque voxel. Dans un voxel se trouvant dans un région impliquée dans le traitement du stimulus visuel présenté, on s'attend à une réponse BOLD, c'est-à-dire une variation dans son décours temporel, suivant le rythme de présentation du stimulus. Pour localiser et créer une carte des régions cérébrales actives, on doit trouver les voxels dont les décours temporels sont reliés à celui du paradigme.

Pour la détection des voxels activés, il existe une pléthore de méthodes disponible. Le plus grand défi réside dans le fort niveau de bruit dans les images à traiter. Une méthode grossière consiste à calculer la moyenne de toutes les images collectées pendant la condition active et à la comparer à la moyenne de toutes les images collectées pendant la condition de repos. Cela devrait révéler où les intensités sont hautes pendant la période active versus la période de contrôle (repos). Des méthodes de détection plus élaborées utilisent en général un modèle temporel ou une série chronologique de référence à laquelle la série temporelle de chacun des voxels est comparée. On procède en général en deux étapes. Pour chaque voxel, une quantité numérique mesurant la proximité entre la série temporelle de référence et celle observée dans le voxel est calculée. On obtient alors une carte de ces valeurs, souvent appelée carte des paramètres statistiques (*Statistical Parameter Map*, SPM). Une haute valeur de proximité pour un voxel signifie qu'il est actif. La carte SPM est par conséquent seuillée afin de classer les voxels dans la catégorie actif/non actif. Elle est ensuite superposée à une image anatomique qui présente une très bonne résolution spatiale (voir la figure 14).

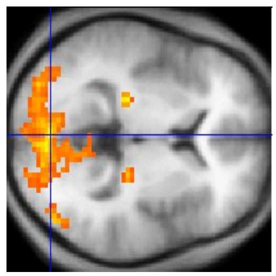


FIG. 14 – Une coupe d'une IRM fonctionnelle du cerveau.

B Résultats mathématiques et algorithmes associés

Résultat 1. *Il est possible d'obtenir les t_m premiers vecteurs propres du cas temporel à partir de ceux du cas spatial.*

Démonstration. La matrice $\dot{X}$ est ici (puisque c'est le cas temporel qui nous intéresse) une matrice de taille $t_m \times v_m$ d'abord centrée en lignes puis en colonnes. D'après la théorie de la décomposition en valeurs singulières (DVS), il est possible de trouver r vecteurs propres orthogonaux $\{\boldsymbol{v}_k\}$ de $\dot{X}^\mathsf{T}\dot{X}$ de taille $v_m \times 1$, et r vecteurs propres orthogonaux $\{\boldsymbol{u}_k\}$ de $\dot{X}\dot{X}^\mathsf{T}$ de taille $t_m \times 1$, où r est le rang de la matrice $\dot{X}$ et $r \leq \min\{v_m, t_m\}$. C'est-à-dire

$$\dot{X}^\mathsf{T}\dot{X}\boldsymbol{v}_k = d_k^2\boldsymbol{v}_k \qquad k = 1, 2, \ldots, r \tag{B.1}$$

$$\dot{X}\dot{X}^\mathsf{T}\boldsymbol{u}_k = d_k^2\boldsymbol{u}_k \qquad k = 1, 2, \ldots, r \tag{B.2}$$

où $\{d_k^2\}$ est l'ensemble des (au plus) r valeurs propres non nulles de $\dot{X}^\mathsf{T}\dot{X}$ (resp. de $\dot{X}\dot{X}^\mathsf{T}$) classées par ordre décroissant.

En multipliant l'équation (B.1) par $\dot{X}$, on peut voir que $\dot{X}\boldsymbol{v}_k$ est un vecteur propre de $\dot{X}\dot{X}^\mathsf{T}$ associé à la valeur propre d_k^2. Donc il s'ensuit que $\boldsymbol{u}_k$ est proportionnel à $\dot{X}\boldsymbol{v}_k$. De la même façon, $\boldsymbol{v}_k$ est proportionnel à $\dot{X}^\mathsf{T}\boldsymbol{u}_k$.

Donc dans le cas temporel, on calculera les t_m valeurs propres $\{d_1^2, \ldots, d_{t_m}^2\}$ et les t_m vecteurs propres $\boldsymbol{u}_k$ de la matrice ${}_{v_m}S_\mathrm{X} = \dot{X}\dot{X}^\mathsf{T}$ (cas spatial), puis on obtiendra les t_m premiers vecteurs propres $\boldsymbol{v}_k$ de ${}_{t_m}S_{\mathrm{X}^\mathsf{T}}$

(cas temporel) en utilisant la formule :

$$\boldsymbol{v}_k = \frac{1}{d_k}\dot{X}^\mathsf{T}\boldsymbol{u}_k, \tag{B.3}$$

soit encore matriciellement

$$\mathcal{V} = \dot{X}^\mathsf{T}\mathcal{U}\mathcal{D}^{-1}. \tag{B.4}$$

Les v_m valeurs propres du cas temporel sont : $\frac{v_m}{t_m}d_1^2, \ldots, \frac{v_m}{t_m}d_{t_m}^2, 0, \ldots, 0$. Les $v_m - t_m$ derniers vecteurs propres du cas temporels ne peuvent être obtenus par cette approche mais ils ne seront pas utiles car ils n'apportent aucune information ($d_i^2 = 0$).

$\square$

Algorithme en R

```
X<-matrix(rnorm(3*5),nrow=3,ncol=5)
tm<-nrow(X)
vm<-ncol(X)
# On traite ici le cas temporel donc on retire la moyenne aux colonnes (variables)
mean.X <- apply(X, 1,mean);Xcentree<-sweep(X, 1, mean.X)
mean.Xcentree <- apply(Xcentree, 2,mean);Xcentree<-sweep(Xcentree, 2, mean.X)
# ou bien Xcentree<-scale(X,scale=F)
svd.Xc.spatial<-svd(Xcentree,nu=tm,nv=0)
U<-svd.Xc.spatial$u
D.moins.1<-diag(1/svd.Xc.spatial$d)
V<-t(Xcentree)%*%U%*%D.moins.1
V[,-ncol(V)]
svd(Xcentree,nu=0,nv=tm-1)$v
```

Résultat 2. *La résolution de l'équation suivante*

$$\hat{\boldsymbol{w}}_j = \underset{\{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{t_m})^\mathsf{T};\|\boldsymbol{\alpha}\|=1\}}{\text{Argmax}} \boldsymbol{\alpha}^\mathsf{T} S_\mathrm{X}\boldsymbol{\alpha} \textit{ avec } \hat{\boldsymbol{w}}_j \textit{ non corrélée à } \hat{\boldsymbol{w}}_1, \ldots, \hat{\boldsymbol{w}}_{j-1}; \quad \forall j = 2, \ldots, t_m. \tag{B.5}$$

nous montre que $\hat{\boldsymbol{w}}_1$ est égal au vecteur propre $\boldsymbol{e}_1$ associé à la plus grande valeur propre λ_1 de la matrice de covariance échantillonnale S_X et que $\mathrm{Var}(\mathrm{Y}_1) = \lambda_1$. De la même façon, on montre que les poids $\hat{\boldsymbol{w}}_2, \ldots, \hat{\boldsymbol{w}}_{t_m}$ des autres composantes sont égaux aux vecteurs propres subséquents $\boldsymbol{e}_2, \ldots, \boldsymbol{e}_{t_m}$ associés aux valeurs propres $\lambda_2, \ldots, \lambda_{t_m}$ de S_X classées par ordre décroissant, avec $\mathrm{Var}(\mathrm{Y}_i) = \lambda_i$, $2 \leq i \leq t_m$.

Résultat 3. *La matrice $\mathcal{E}$ contenant les vecteurs propres de la matrice de covariance échantillonnale S_X est une matrice orthogonale ($\mathcal{E}\mathcal{E}^\mathsf{T} = \mathcal{E}^\mathsf{T}\mathcal{E} = \mathcal{I}$) et il s'agit donc d'une matrice de rotation.*

C Quelques algorithmes en R

Algorithme de résolution de l'ACPs

```
# X est une matrice de taille tm x vm qui contient les données en colonnes (tm<<vm)
load("bonrespoursection8.R") # on charge un jeu de données
X<-res$donnees
tm<-dim(X)[1]
vm<-dim(X)[2]
Xc<-t(scale(t(X),scale=F)) # on centre les données en lignes
image(cor(t(X))) # pour regarder les corrélations
propre<-eigen(cov(t(X))*(vm-1)/vm) # calcul des valeurs et vecteurs propres de S_x
# pour regarder l'éboulis des valeurs propres
plot(sort(propre$values,decreasing=T),type="b",cex=.2,ylab="Valeur propre")
Y<-t(propre$vectors[,c(1,2)])%*%Xc # calcul des 2 premières composantes principales
plot(t(Y),xlab="C1",ylab="C2") # on trace les points projetés
abline(h=0,v=0) # et les axes du repère

# on peut utiliser svd à la place de eigen
svd.Xc<-svd(Xc)
svd.Xc$d^2 # identique à propre$values*vm
svd.Xc$u # identique à propre$vectors (au signe près des colonnes)
```

Algorithme de résolution de l'ACPt

```
# Xt est une matrice de taille vm x tm qui contient les données en colonnes (tm<<vm)
load("bonrespoursection8.R") # on charge un jeu de données
Xt<-t(res$donnees)
tm<-dim(Xt)[2]
vm<-dim(Xt)[1]
Xtc<-t(scale(t(Xt),scale=F)) # on centre les données en lignes
image(cor(t(Xt))) # pour regarder les corrélations
propreSxt<-eigen(cov(t(Xt))*(tm-1)/tm) # calcul des valeurs et vecteurs propres de Sxt
# pour regarder l'éboulis des valeurs propres
plot(sort(propreSxt$values,decreasing=T),type="b",cex=.2,ylab="Valeur propre")
Y1<-t(propreSxt$vectors[,c(1)])%*%Xtc # calcul de la première composante principale
Y2<-t(propreSxt$vectors[,c(2)])%*%Xtc # calcul de la deuxième composante principale
par(mfrow=c(2,1)) # on divise la fenêtre graphique
plot(t(Y1),type="l") # on trace les points projetés
plot(t(Y2),type="l") # on trace les points projetés

# on peut aussi utiliser la DVS complète
svd.Xtc<-svd(Xtc,nu=vm,nv=tm)
c(svd.Xtc$d,rep(0,vm-tm))^2 # identique à propreSxt$values*tm
svd.Xtc$u # c'est V_f # identique à propreSxt$vectors (au signe près des colonnes)


# ou bien la DVS compacte, très utile pour limiter le temps de calcul
svd(Xtc,nu=tm,nv=tm)$d^2 # identique à (tm*propreSxt$values[1:tm]) si tm<vm
svd(Xtc,nu=tm,nv=tm)$u # identique à (propreSxt$vectors[,1:tm]) (au signe près des colonnes)

# ou encore la DVS tronquée
```

```
# on approche la matrice en utilisant tmmin < tm
# tmmin peut être choisi en éliminant les petites valeurs singulières
tmmin<-sum(svd(Xtc,nu=0,nv=0)$d>0.1)
# Comparer Xtcred avec Xtc
Xtcred<-svd(Xtc,nu=tmmin,nv=tmmin)$u%*%diag(svd(Xtc,nu=tmmin,nv=tmmin)$d[1:tmmin])%*%
          t(svd(Xtc,nu=tmmin,nv=tmmin)$v)
```

Algorithme ACIs après ACPs

```
varpop<-function(X){var(t(X))*(nrow(t(X))-1)/nrow(t(X))}

X<-matrix(rnorm(3*5),nrow=3,ncol=5)
tm<-nrow(X)
vm<-ncol(X)
mean.X <- apply(X, 1,mean);Xcentree<-sweep(X, 1, mean.X)
# ou bien Xcentree<-t(scale(t(X),scale=F))


U<-svd(Xcentree,nv=0)$u
D<-diag(svd(Xcentree,nu=0,nv=0)$d)

s<-La.svd(Xcentree,nu=0)

Z<-sqrt(vm)*s$v
# On peut vérifier que varpop(Z)=Id

n.comp <- min(vm, tm)
w.init <- matrix(rnorm(n.comp^2), n.comp, n.comp)
maxit<-200
tol<-1e-04
verbose<-FALSE
alpha<-1
fun = c("logcosh")
require(fastICA)
#le prog ica.R.def renvoie une estimation de la matrice de séparation
Wz <- ica.R.def(Z, n.comp, tol = tol, fun = fun,alpha = alpha, maxit = maxit,
verbose = verbose,w.init = w.init)
Ax <- U%*%D%*%solve(Wz)/sqrt(vm)
Wx <- sqrt(vm)*Wz%*%diag(1/diag(D))%*%t(U)
Sest<-Wz%*%Z
# ou bien: Sest<-Wx%*%Xcentree


# Equivalent : res$A==t(Ax) et res$W==t(Wz)
res <- fastICA(t(X), n.comp=tm, alg.typ = "deflation",fun = "logcosh", alpha = 1, method = "R",
row.norm = FALSE, maxit = 200, tol = 1e-04, verbose = FALSE,
             w.init = w.init)



# On a Xc==Xcentree
Xc<-matrix(0,nrow=tm,ncol=vm)
```

```
for (j in 1:tm) {
Xc<-Xc+outer(Ax[,j],Sest[j,])
}

# On choisit en général r en examinant les valeurs propres diag(D^2)
# et en ne retenant que les plus grandes
r<-2
Ured<-svd(Xcentree,nu=r,nv=0)$u
Dred<-diag(svd(Xcentree,nu=0,nv=0)$d[1:r])

sred<-La.svd(Xcentree,nu=0,nv=r)

Zred<-sqrt(vm)*sred$v
# On peut vérifier que varpop(Zred)==Id

n.comp <- r
w.init <- matrix(rnorm(n.comp^2), n.comp, n.comp)
maxit<-200
tol<-1e-04
verbose<-FALSE
alpha<-1
fun = c("logcosh")
require(fastICA)
#le prog ica.R.def renvoie une estimation de la matrice de séparation
Wzred <- ica.R.def(Zred, n.comp, tol = tol, fun = fun,alpha = alpha,
maxit = maxit, verbose = verbose,w.init = w.init)
Wxred <- sqrt(vm)*Wzred%*%diag(1/diag(Dred))%*%t(Ured)
Axred <- Ured%*%Dred%*%solve(Wzred)/sqrt(vm)
# ou bien Axred<-t(Wxred)%*%solve(Wxred%*%t(Wxred))
Sredest<-Wzred%*%Zred
# ou bien: Sredest<-Wxred%*%Xcentree


# Equivalent : resred$A==t(Axred) et resred$W==t(Wzred)
resred <- fastICA(t(X), n.comp=r, alg.typ = "deflation",fun = "logcosh", alpha = 1, method = "R",
row.norm = FALSE, maxit = 200, tol = 1e-04, verbose = FALSE,
             w.init = w.init)

# On a Xcred==Axred%*%Sredest \approx Xcentree
Xcred<-matrix(0,nrow=tm,ncol=vm)
for (j in 1:r) {
Xcred<-Xcred+outer(Axred[,j],Sredest[j,])
}
```

Algorithme ACIt après ACPt

```
varpop<-function(X){var(t(X))*(nrow(t(X))-1)/nrow(t(X))}

X<-matrix(rnorm(3*7),nrow=3,ncol=7)
Xt<-t(X)
tm<-nrow(X)
vm<-ncol(X)
```

```
mean.Xt <- apply(Xt, 1,mean);Xtcentree<-sweep(Xt, 1, mean.Xt)
# ou bien Xtcentree<-t(scale(t(Xt),scale=F))

D<-diag(svd(Xtcentree,nu=0,nv=0)$d[1:(tm-1)])
Ut<-La.svd(Xtcentree,nu=0)$v[1:(tm-1),]
V<-svd(Xtcentree,nv=0)$u[,1:(tm-1)]

Z<-sqrt(tm)*Ut

# On peut vérifier que varpop(Z)=Id

n.comp <- tm-1
w.init <- matrix(rnorm(n.comp^2), n.comp, n.comp)
maxit<-200
tol<-1e-04
verbose<-FALSE
alpha<-1
fun = c("logcosh")
require(fastICA)
#le prog ica.R.def renvoie une estimation de la matrice de séparation
Wz<-ica.R.def(Z, n.comp, tol = tol, fun = fun,alpha = alpha, maxit = maxit,
verbose = verbose,w.init = w.init)

Wz

# PROBLEME, C'EST DIFFERENT DE (probablement à cause de l'initialistion):
Wz/ica.R.def(diag(c(1,-1))%*%Z, n.comp, tol = tol, fun = fun,alpha = alpha, maxit = maxit,
verbose = verbose,w.init = w.init)

Axt <- V%*%D%*%solve(Wz)/sqrt(tm)
Wxt <- sqrt(tm)*Wz%*%diag(1/diag(D))%*%t(V)
Sest<-Wz%*%Z
# ou bien: Sest<-Wxt%*%Xtcentree


# Equivalent : res$A==t(Axt) et res$W==t(Wz)
res <- fastICA(t(Xt), n.comp=tm-1, alg.typ = "deflation",fun = "logcosh", alpha = 1, method = "R",
row.norm = FALSE, maxit = 200, tol = 1e-04, verbose = FALSE,
             w.init = w.init)



# On a Xtc==Xtcentree
Xtc<-matrix(0,nrow=vm,ncol=tm)
for (j in 1:(tm-1)) {
Xtc<-Xtc+outer(Axt[,j],Sest[j,])
}

# On choisit en général r en examinant les valeurs propres diag(D^2)
# et en ne retenant que les plus grandes
r<-2
```

```
Ured<-svd(Xcentree,nu=r,nv=0)$u
Dred<-diag(svd(Xcentree,nu=0,nv=0)$d[1:r])

sred<-La.svd(Xcentree,nu=0,nv=r)

Zred<-sqrt(vm)*s$v
# On peut vérifier que varpop(Zred)==Id

n.comp <- r
w.init <- matrix(rnorm(n.comp^2), n.comp, n.comp)
maxit<-200
tol<-1e-04
verbose<-FALSE
alpha<-1
fun = c("logcosh")
require(fastICA)
#le prog ica.R.def renvoie une estimation de la matrice de séparation
Wzred <- ica.R.def(Zred, n.comp, tol = tol, fun = fun,alpha = alpha, maxit = maxit,
verbose = verbose,w.init = w.init)
Wxred <- sqrt(vm)*Wzred%*%diag(1/diag(Dred))%*%t(Ured)
Axred <- Ured%*%Dred%*%solve(Wzred)/sqrt(vm)
# ou bien Axred<-t(Wxred)%*%solve(Wxred%*%t(Wxred))
Sredest<-Wzred%*%Zred
# ou bien: Sredest<-Wxred%*%Xcentree


# Equivalent : resred$A==t(Axred) et resred$W==t(Wzred)
resred <- fastICA(t(X), n.comp=r, alg.typ = "deflation",fun = "logcosh", alpha = 1, method = "R",
row.norm = FALSE, maxit = 200, tol = 1e-04, verbose = FALSE,
             w.init = w.init)

# On a Xcred==Axred%*%Sredest \approx Xcentree
Xcred<-matrix(0,nrow=tm,ncol=vm)
for (j in 1:r) {
Xcred<-Xcred+outer(Axred[,j],Sredest[j,])
}
```

D Le simulateur de signaux IRMF cérébraux

Afin de pouvoir évaluer quantitativement les méthodes d'analyse statistique décrites dans ce mémoire, nous avons créé un outil informatique (voir [LSA$^+$99], [RM04] et [FBLK02] pour des outils similaires) permettant de générer un signal temporel dans un ensemble de pixels. Une utilisation très simplifiée de ce programme consiste à délimiter deux zones dans une image carrée constituée de 15×15 cases. Les positions de ces deux zones sont choisies aléatoirement dans l'image. La première zone A est un motif quelconque de 30 cases voisines, la deuxième zone B est rectangulaire de taille 4×2. Dans la zone A, on génère un signal temporel $x_A(t)$. Dans la zone B, on génère un signal temporel $x_B(t)$. Enfin, on superpose dans toutes les cases un signal correspondant à un bruit gaussien. Les signaux sont de longueur 216.

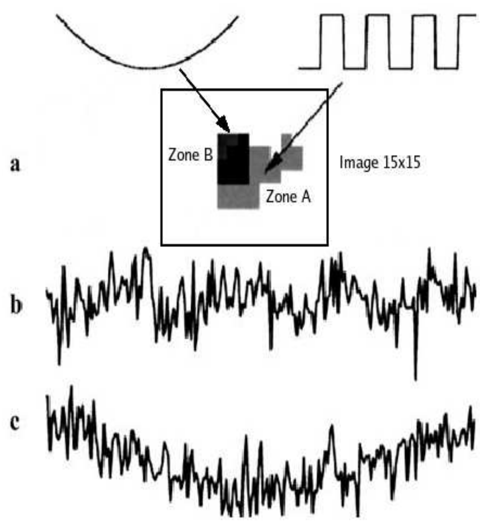


FIG. 15 – (a) Une image de 15×15 cases contient deux zones. Dans la zone A, on génère un signal créneau. Dans la zone B, on génère un signal quadratique. On ajoute dans toutes les cases un bruit gaussien. (b) Exemple de décours temporel généré dans la zone A. (c) Exemple de décours temporel généré dans la zone B.

Les signaux générés dans les zones A et B sont en fait préalablement convolués par un modèle de la fonction de réponse hémodynamique (HRF) afin de mieux approcher la réalité des signaux effectivement mesurés par l'IRMF. Les figures 16 et 17 montrent l'effet de cette convolution sur les signaux $x_A(t)$ et $x_B(t)$.

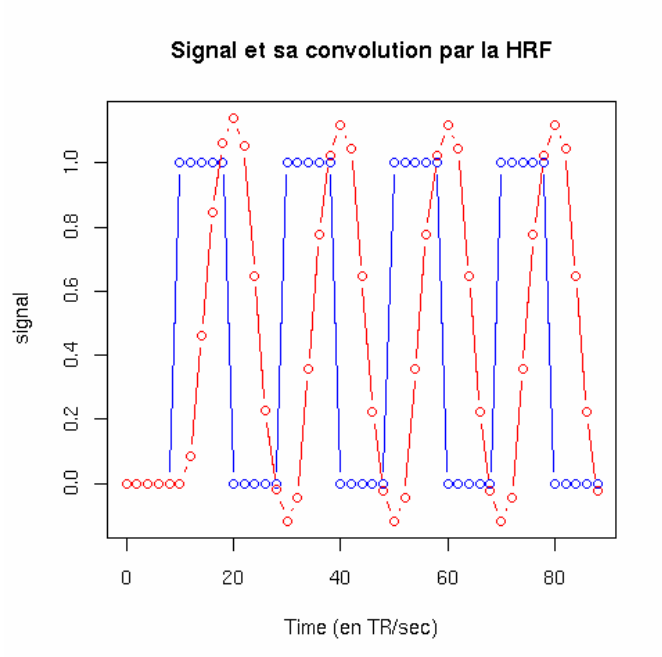


FIG. 16 – Signal $x_A(t)$ et le résultat de sa convolution avec la HRF.

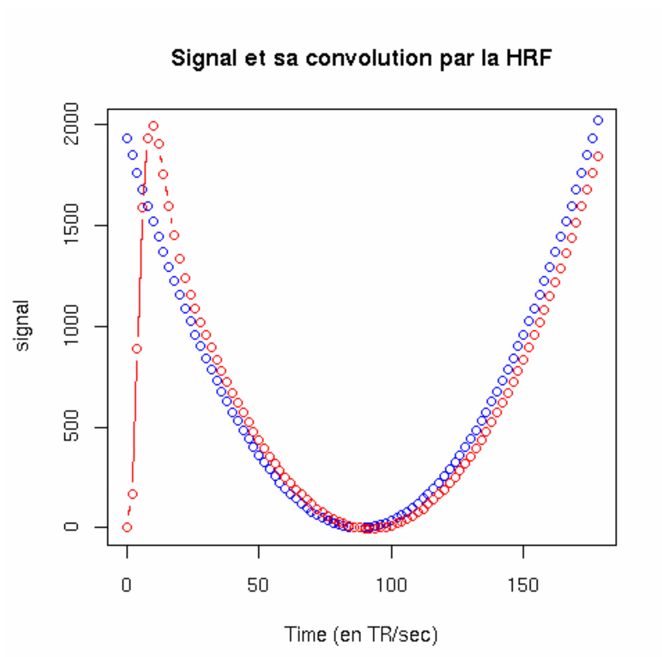


FIG. 17 – Signal $x_B(t)$ et le résultat de sa convolution avec la HRF.

Voilà un exemple du style de données simulées par notre programme parmi lesquelles il s'agit de retrouver les zones spatiales A et B d'origine ainsi que les décours temporels qui y ont été générés.

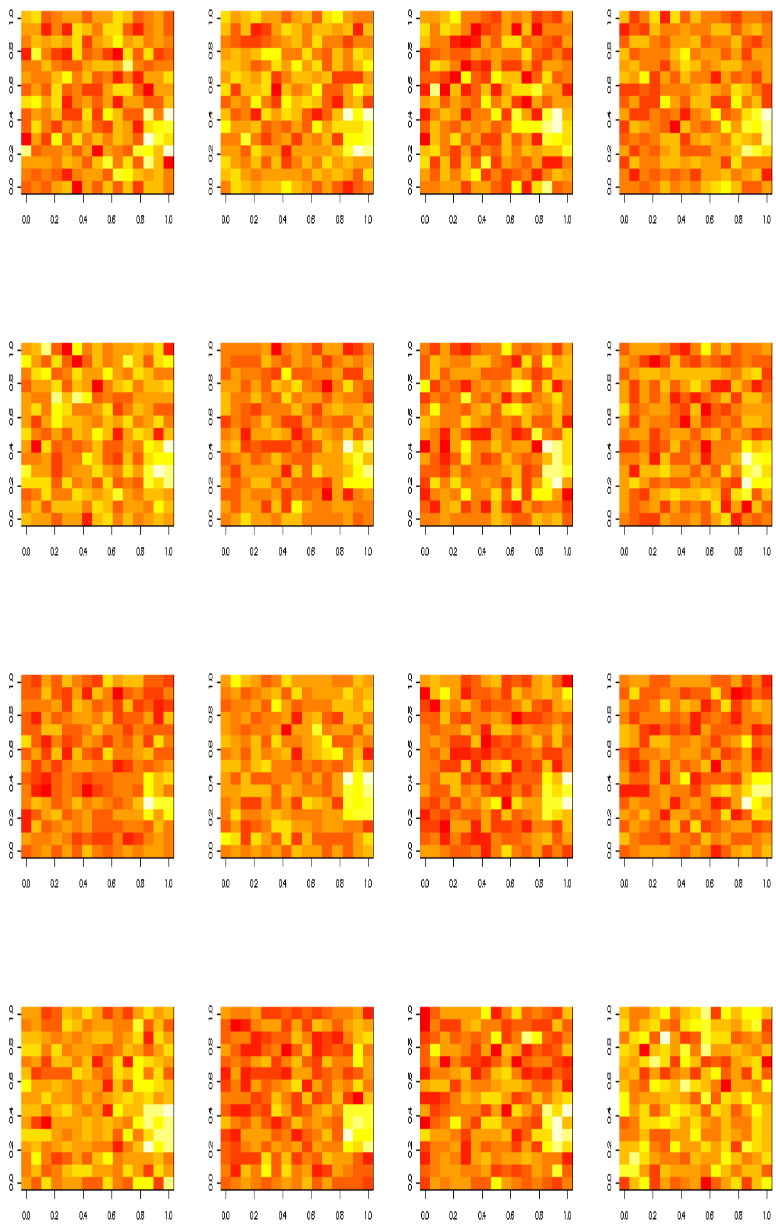


FIG. 18 – Petit échantillon de données simulées

Nous appliquons ensuite les algorithmes d'ACP et d'ACI dans leur version spatiale et temporelle. Les méthodes spatiales cherchent à retrouver les deux motifs spatiaux (et leur décours temporel associé) alors que les méthodes temporelles cherchent à retrouver les signaux temporels quadratique et créneau (et les zones où ils sont localisés).

Il apparaît nettement dans les résultats s'affichant sur les graphiques des figures 19, 20, 21 et 22 que ces méthodes statistiques parviennent à retrouver les motifs spatiaux et les signaux temporels sources.

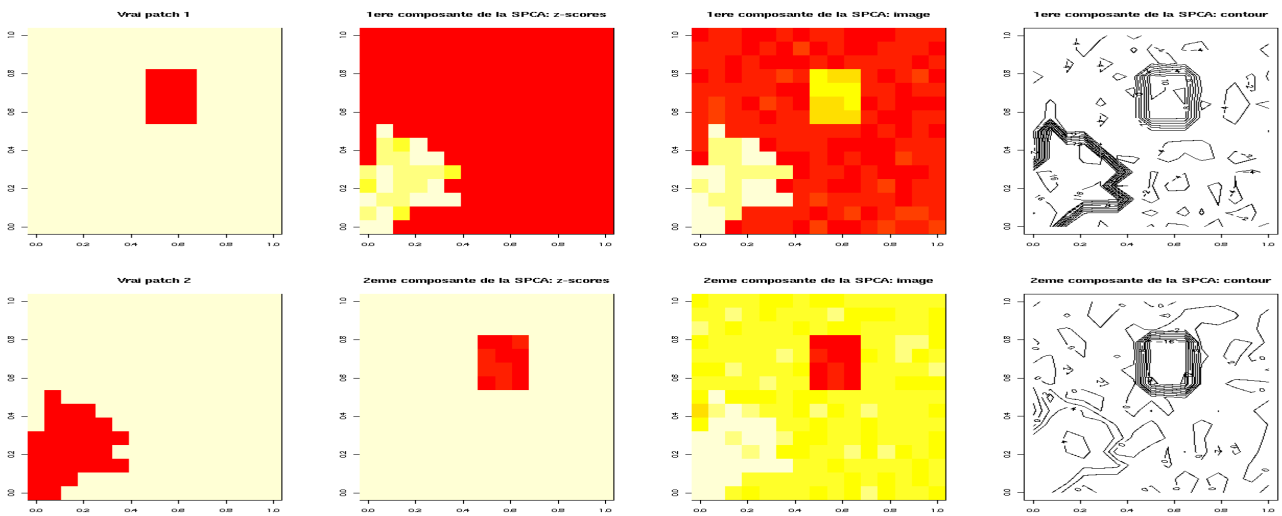


FIG. 19 – Motifs spatiaux originaux et estimés par l'ACPs

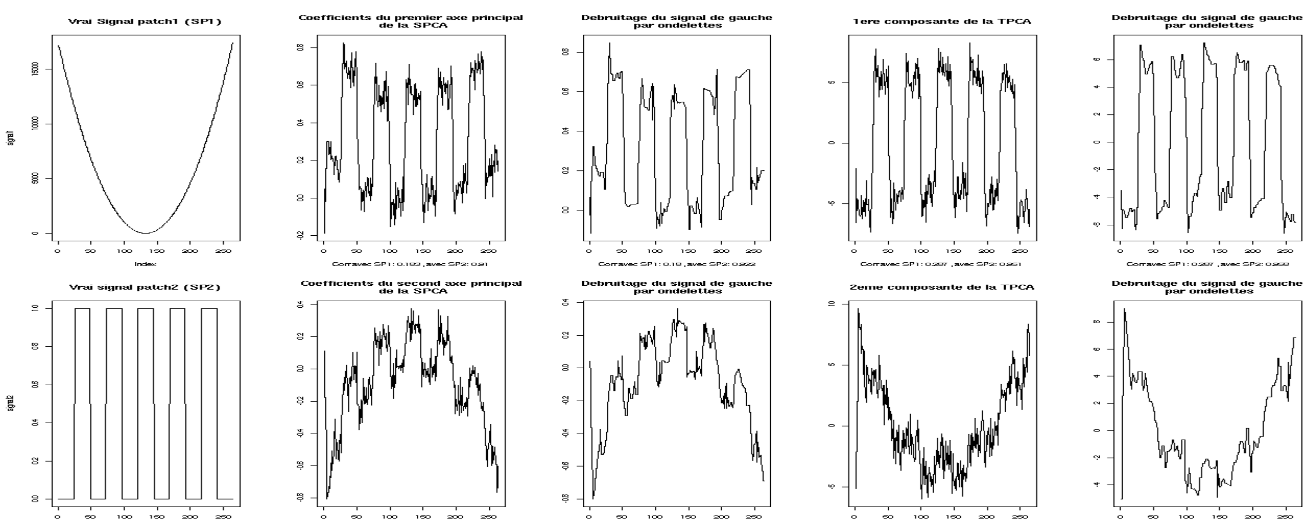


FIG. 20 – Décours temporels originaux et estimés par l'ACPt

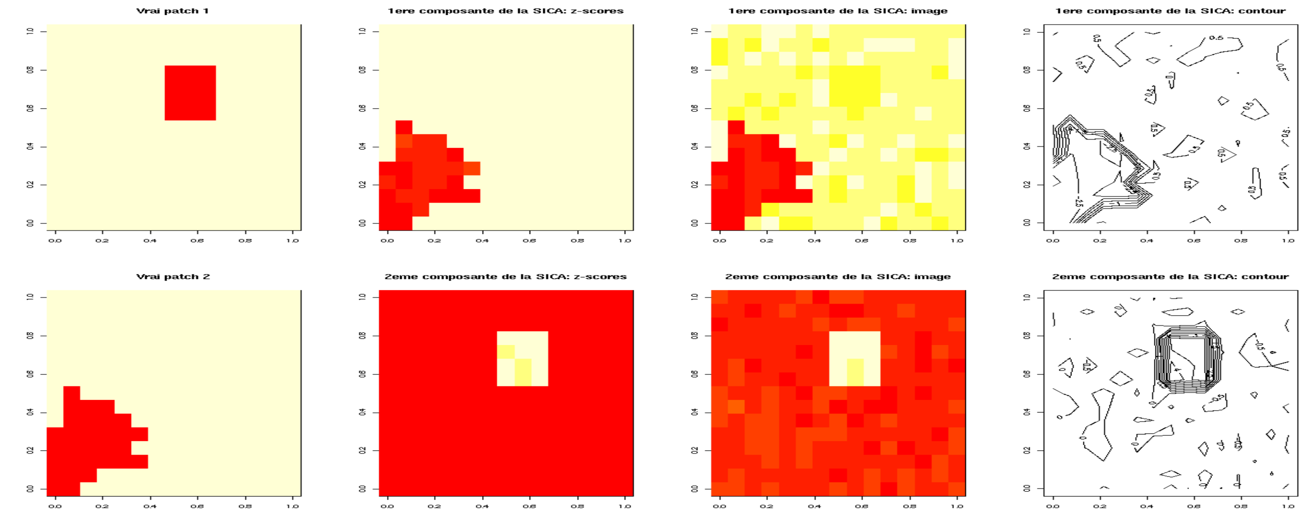


FIG. 21 – Motifs spatiaux originaux et estimés par l'ACIs

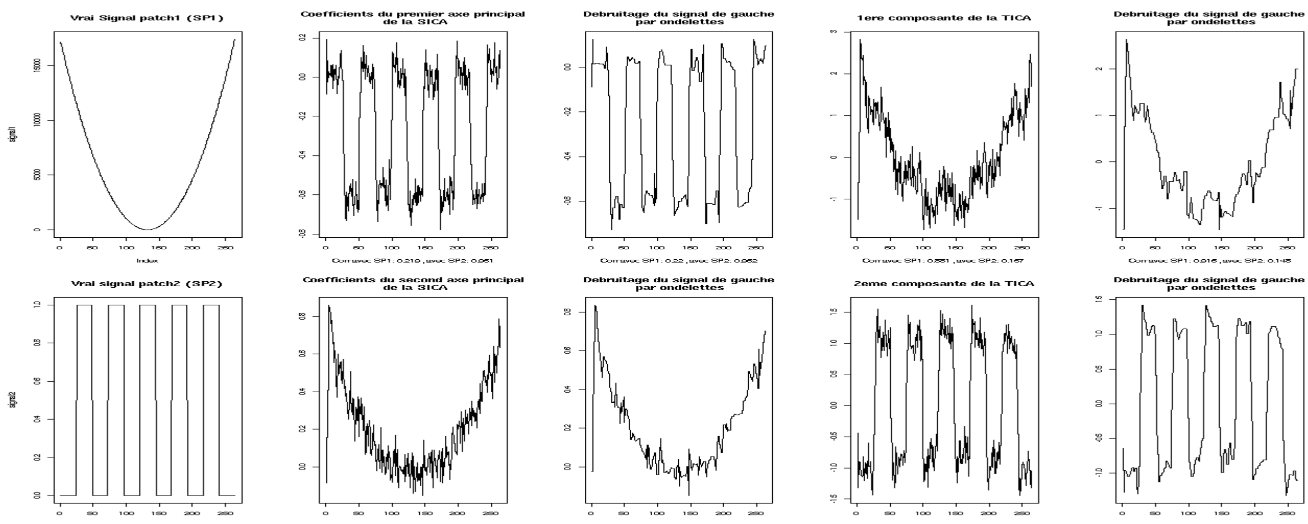


FIG. 22 – Décours temporels originaux et estimés par l'ACIt

Dans la pratique, toutefois, un grand nombre de signaux temporels seront extraits et le problème ensuite sera de retrouver la composante, parmi toutes celles obtenues, qui correspond au stimulus. Pour cela, on peut essayer de trouver le décours temporel extrait qui est le plus corrélé avec le signal temporel du stimulus (convolué avec la HRF). Une autre idée est d'utiliser à la place de la corrélation des mesures de dépendance plus sophistiquées : cross-entropie, cross-redondance, etc ... (voir [GMR04], [DR03], [Gro97], [HHS06], [KL05], [PS96], [PMI$^+$96], [PT95] et [RM00])

Un autre problème est de choisir dans la carte spatiale associée au décours temporel sélectionné les voxels qui peuvent être considérés comme significativement actifs.

Voilà une première façon de procéder. Notons A_{jl} la valeur du voxel l de la composante (IC) numéro j.
On a $x_{tl} = d_{t1}A_{1l} + d_{t2}A_{2l} + \ldots + d_{tK}A_{Kl}$ où K est le nombre de compoantes.
Soit encore $x_{tl} = \sum_{j=1}^{K} d_{tj}a_{jl}$.
Ainsi, par exemple, la contribution (au signal obervé $(x_{tl})_{t=1,\ldots,t_m}$ dans le voxel numéro l) de la 1ère composante IC1 est $(d_{11},\ldots,d_{T1})A_{1l}$.
La source 1 (IC1) s'exprime donc, suivant le décours temporel $(d_{11},\ldots,d_{T1})$, avec un poids A_{1l} dans le voxel numéro l.
Le voxel numéro l sera dit actif pour la composante (la source) IC1 si le poids A_{1l} est différent de 0 (en moyenne). Pour cela, on peut faire un test d'hypothèse : $H_0 : \mathbf{E}[\mathrm{A}_{1l}] = 0$ versus $H_1 : \mathbf{E}[\mathrm{A}_{1l}] \neq 0$.
Si l'on suppose que la variable aléatoire A_{1l} (dont la réalisation est A_{1l}) suit une loi normale $N(\mu_l, \sigma^2)$ pour tout l, alors on a $\frac{\mathrm{A}_{1l}-\mu_l}{\sigma} \sim N(0,1)$ et sous $H_0 : \mu_l = 0$ (= pas d'activation du voxel l), on a $\frac{\mathrm{A}_{1l}}{\sigma} \sim N(0,1)$ et $\frac{\mathrm{A}_{1l}}{\hat{\sigma}} \sim \mathcal{T}(v_m - 1)$, loi de student à $v_m - 1$ degrés de liberté où v_m est le nombre de voxels et avec $\hat{\sigma} = \frac{1}{v_m - 1}\sum_{l=1}^{v_m}(\mathrm{A}_{1l} - \hat{\mu})^2$ et $\hat{\mu} = \frac{1}{v_m}\sum_{l=1}^{v_m}\mathrm{A}_{1l}$.

Une autre possibilité est d'appliquer des méthodes Bootstrap : simuler avec remise $(A_{1l})_b^*, b = 1,\ldots,B$ et puis faire un test-t sur ces valeurs.

Une autre piste intéressante à explorer serait de garder uniquement les voxels ayant un fort cos^2 comme cela se fait souvent en ACP.

E Application sur un jeu de données réelles

Les données que nous avons utilisées dans ce travail proviennent de l'étude en cours menée par M. Dojat (GIN, Grenoble) et J.M. Hupé (CERCO, Toulouse) sur la perception des couleurs synesthésiques[1]. Il s'agit d'identifier les structures cérébrales impliquées lors de cette perception visuelle à l'aide de l'imagerie médicale par résonance magnétique fonctionnelle et de comparer avec le réseau impliqué dans la perception de couleurs "réelles" pour des synesthètes et des non-synesthètes. Un des buts de cette étude est de mieux comprendre l'implication de l'aire V4 largement décrite dans la littérature comme une zone centrale dans la perception colorée. Une session est réalisée pour délinéer l'aire V4. Ce sont ces données que nous avons utilisées.

Protocole de délinéation de V4 :
Le protocole alterne des blocs de 12.5 sec. où sont présentés successivement des figures géométriques colorées, les mêmes figures achromatiques et un point de fixation. Durant chaque bloc, 5 images sont acquises (TR= 2.5 sec.). Il y a au total 8 répétitions soit 120 images. Pour le sujet map184 cette session a été répétée mais en présentant d'abord les images achromatiques (soit 240 images acquises). La comparaison des blocs figures (colorées ou achromatiques) versus le repos reflète une activité dans toutes les aires visuelles. La comparaison des blocs chromatiques versus achromatiques permet de localiser l'aire V4.

Acquisition :
Les acquisitions d'images sont réalisées sur l'IRM 3T de l'IFR 1 (Grenoble, Fr). Les stimuli seront générés avec un PC standard, le langage Matlab (The Mathworks Inc), Visual C++ (Microsoft Inc) et le logiciel Presentation (Neurobehavioral Systems Inc).

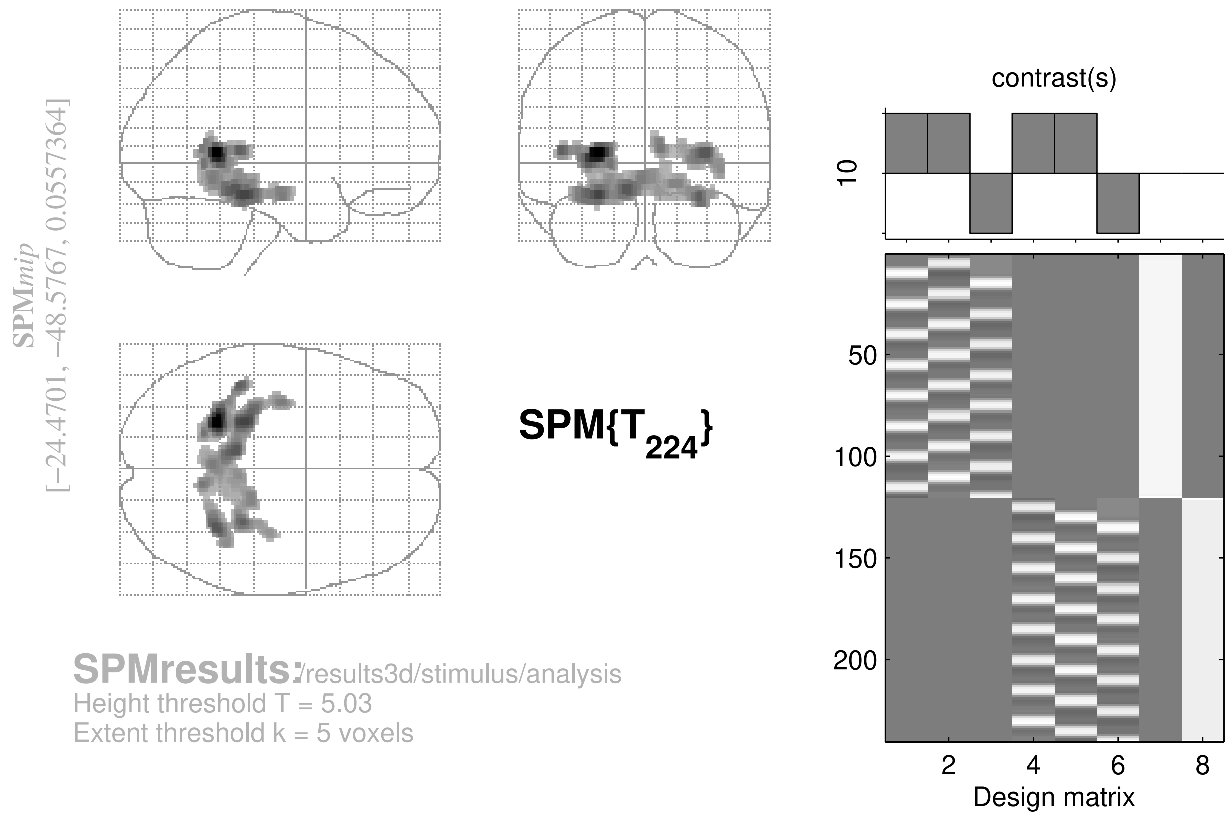
- *Installation dans l'aimant*. Le sujet s'installe sur le lit d'examen, où on lui fixe la tête à l'aide des cales synthétiques. Nous utilisons une antenne tête afin de privilégier l'acquisition d'une bonne image anatomique, nécessaire pour les procédures de segmentation et de dépliage du cortex. Pour les images fonctionnelles, cela permet d'observer si l'expérience de synesthésie génère des activités également en dehors du cortex visuel. Un miroir est fixé au niveau des yeux du sujet afin de lui permettre de voir les images.
- *Acquisition préliminaire*. Après l'entrée du sujet au centre de la machine, une première acquisition de repérage est effectuée avec l'antenne tête. Elle utilise une séquence 3D en écho de gradient, pondérée T1, qui fournit des images anatomiques à partir desquelles les coupes des acquisitions suivantes seront positionnées (durée 4').
- *Acquisitions pour l'expérience de localisation de V4*. Après le repérage, chaque expérience comprendra l'acquisition de 120 images (TR = 2.5, durée 6min, résolution $3 \times 3 \times 3$ mm^3). Une anatomique 3D (4') est acquise (résolution 1mm^3) à la fin ainsi que deux images pour établir la carte spatiale du champ magnétique en vue de la correction des distorsions géométriques.

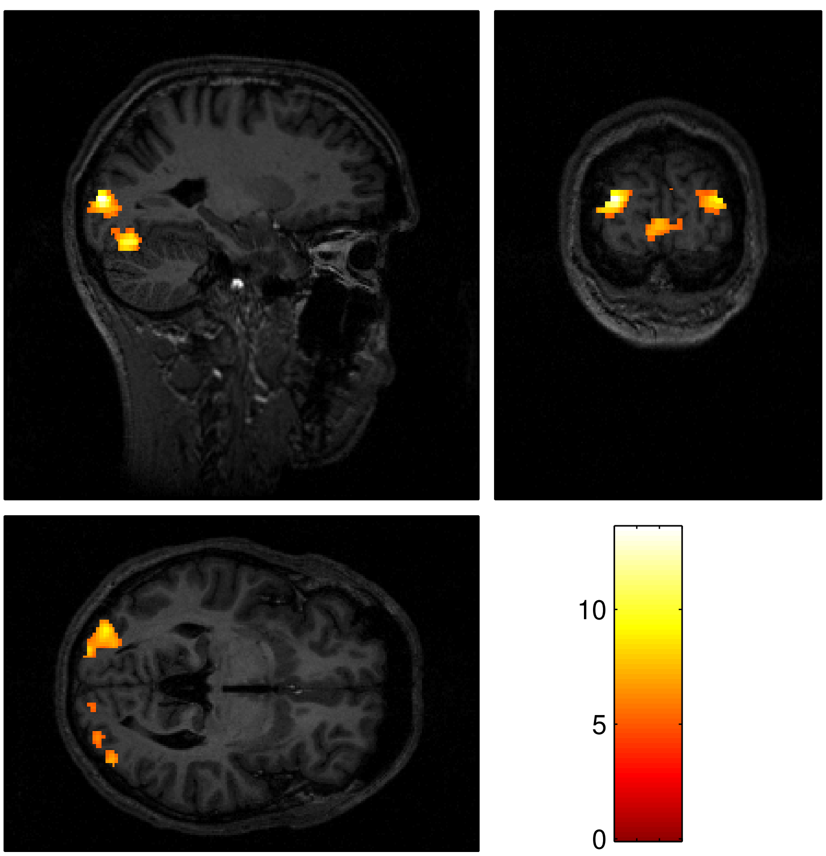
Préparation des données :
Après conversion au format Amalyze (SPM2), les données fonctionnelles sont réalignées (correction du mouvement) sur une référence (la fonctionnelle la plus proche de l'anatomique) et corrigées des distorsions géométriques. L'anatomique est ensuite réalignée sur l'image fonctionnelle moyenne. Un lissage spatial de $6 \times 6 \times 6$ mm^3 est ensuite appliqué aux images fonctionnelles. Les données ont ensuite été analysées à l'aide d'un modèle linéaire généralisé à l'aide du logiciel SPM2. Les cartes statistiques représentatives du contraste Vision versus Repos et Couleur versus Achromatique ont été obtenues pour comparaison avec l'analyse par l'ACI.

Voilà les résultats obtenus à l'aide du logiciel SPM2 par M. Dojat.

[1]Les synesthésies, ou "union des sens", décrivent les particularités, non pathologiques, partagées par des personnes pour qui la stimulation dans une modalité sensorielle provoque de façon systématique une sensation dans une autre modalité sensorielle, comme dans le cas de l'audition colorée, où des sons produisent des sensations de couleur. On parle également de synesthésie pour l'association de couleurs ou de personnalités à des chiffres et des lettres.

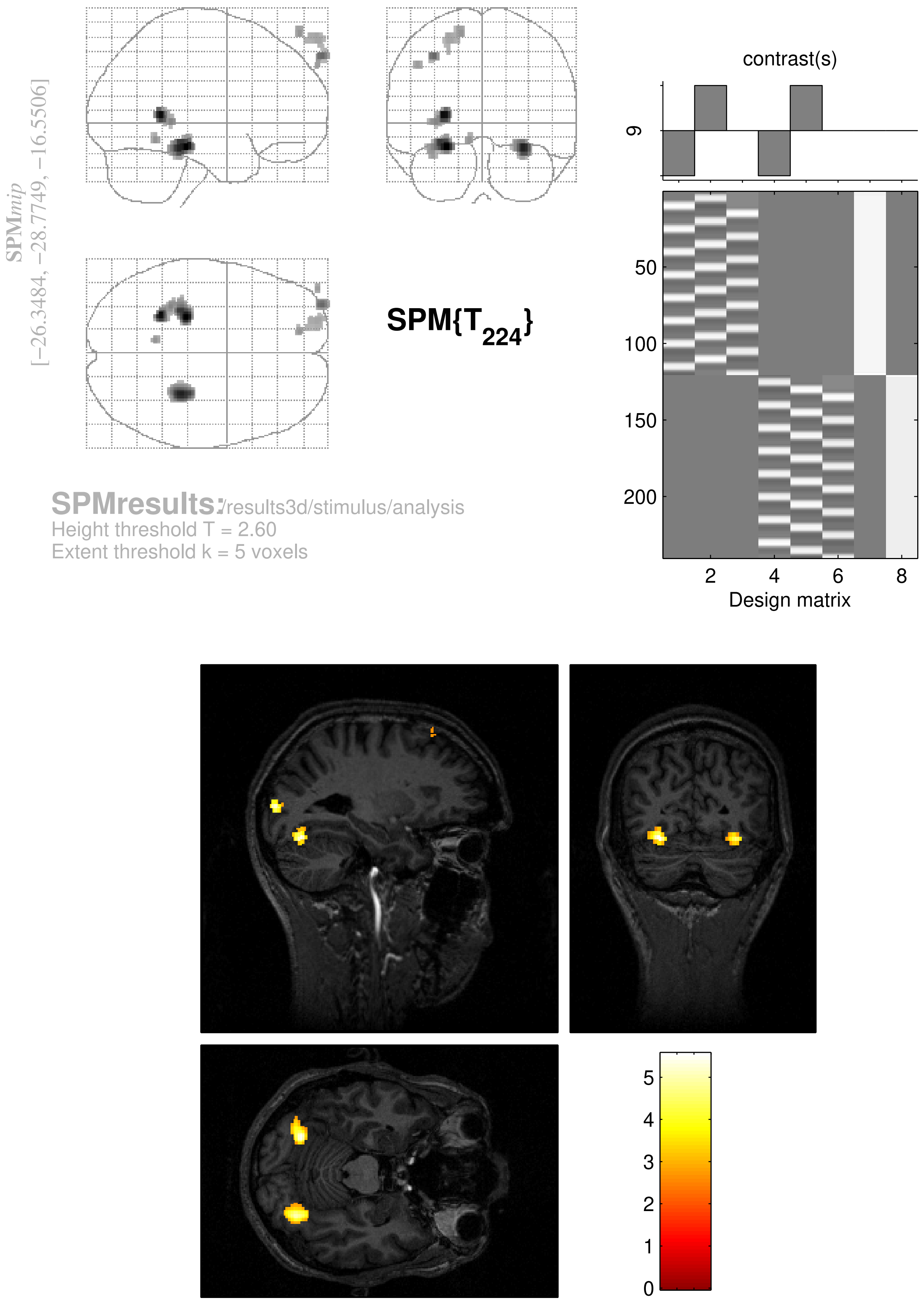
differential effects Visual vs Rest





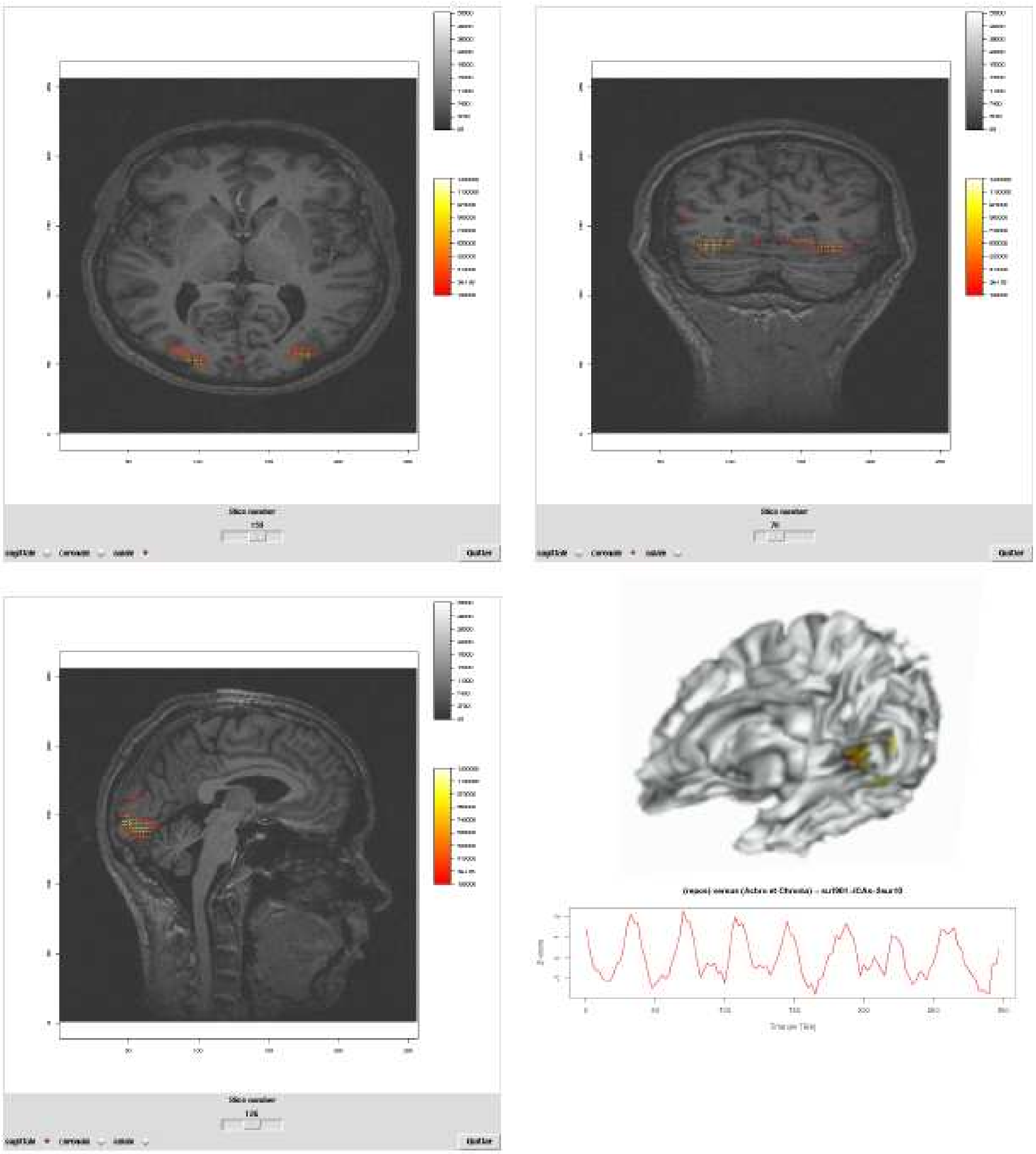
SPM2 (mdojat): 16:35:10 – 01/06/2007

differential effects Color vs Achro

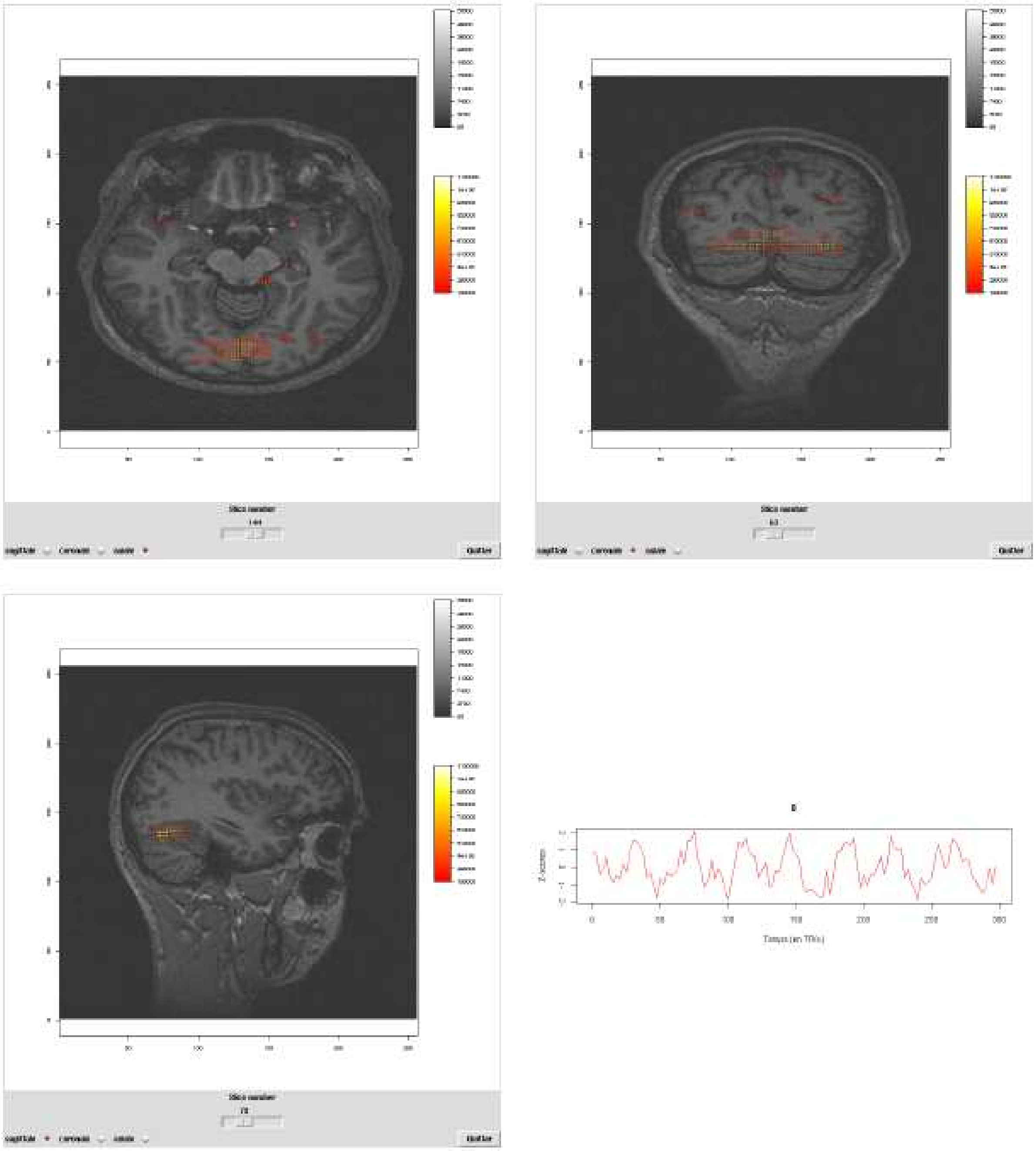


SPM2 (mdojat): 16:39:06 – 01/06/2007

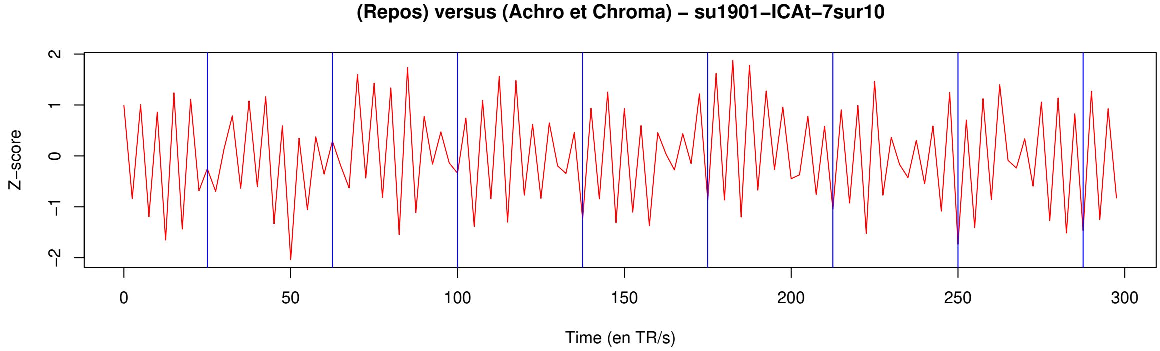
Nous présentons maintenant les résultats obtenus en appliquant une ACI spatiale sur le sujet map284-1901 (une seule session). Ces résultats et leur visualisation ont été obtenus à l'aide du logiciel R. Le deuxième décours temporel sur les 10 calculés (celui qui est le plus corrélé avec le signal du paradigme convolué avec la HRF : corrélation=0.815) est présenté en bas à droite. C'est celui qui est associé à la composante indépendante spatiale seuillée (p=0.97) obtenue par l'ACIs et présentée dans les trois autres cadres de l'image (en coupe axiale, coronale et sagittale, de gauche à droite et de haut en bas). Les activations dans le cortex visuel sont nettement visibles et l'on retrouve sensiblement les mêmes zones que celles obtenues avec le logiciel SPM. Nous avons aussi rajouté une vision en 3D (obtenue à l'aide du logiciel brainvisa) de la projection de cette activation sur la surface du cortex d'un demi-hémisphère.



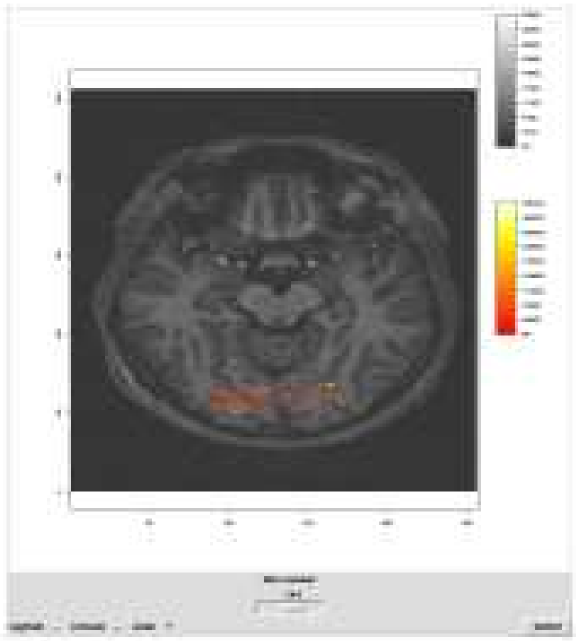
Et voici deux décours temporels intéressants obtenus par application de l'ACI temporelle. On peut constater que le premier est bien corrélé avec le signal du paradigme (corrélation=0.68) et qu'il se situe bien dans le cortex visuel.

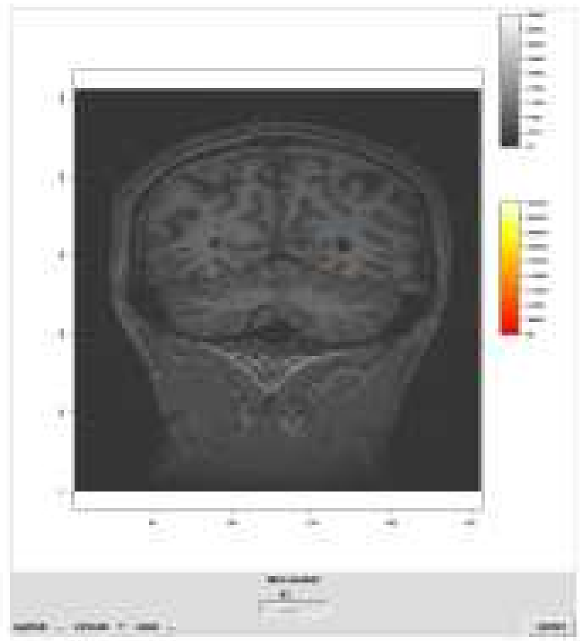


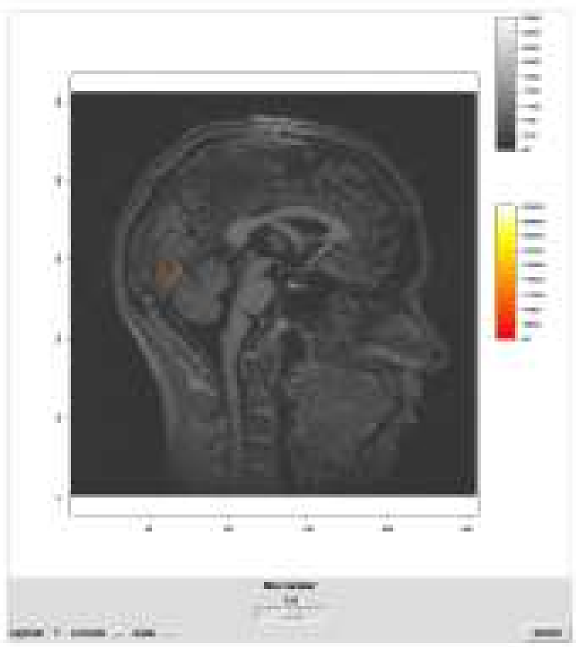
Le second signal présente une structure très intéressante mais nous n'avons pour l'instant pas pu l'expliquer. Il se situe principalement dans le cortex visuel, le corps calleux et le cervelet.



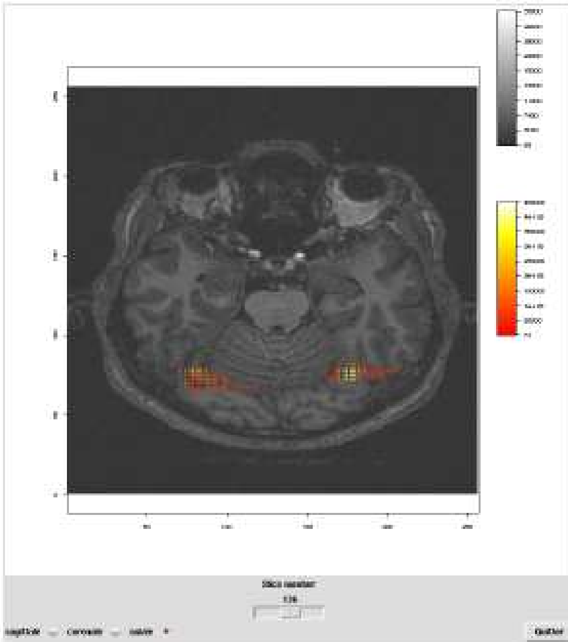
Une idée intéressante a consisté à effectuer d'abord une ACI spatiale, ce qui a permis de sélectionner une zone de petite taille (5420 voxels) dans laquelle on pouvait penser que le signal du paradigme s'exprimait. On a ensuite appliqué une ACI temporelle uniquement sur les données de cette nouvelle zone.
Voilà les deux décours temporels intéressants que nous avons trouvé :

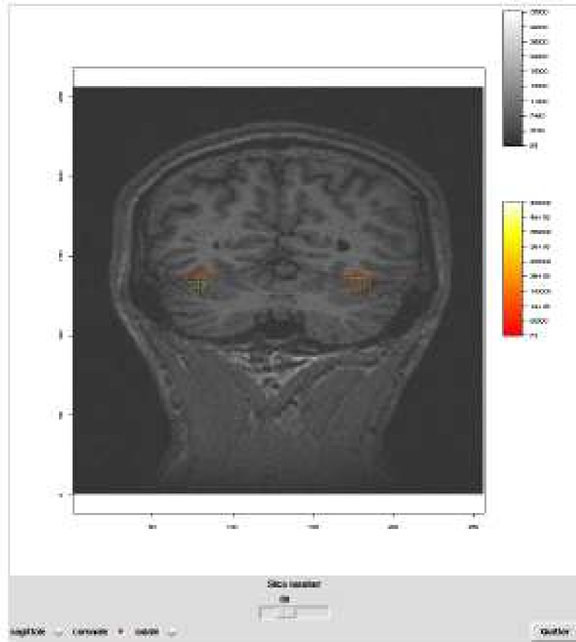


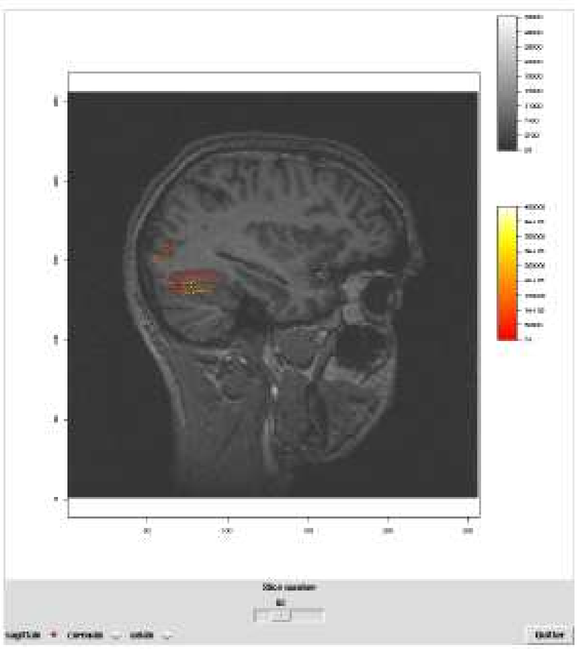


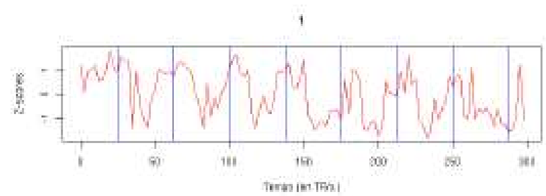


Nous voyons ici nettement les zones (V4) qui ne sont sensibles qu'à la couleur.









F Programmes informatiques en R

Voici le code source R de l'ensemble des programmes informatiques développés pendant ce stage.

F.1 Fonction effectuefastica

```
source("brainsimul.R")
source("signal-temporel.R")
source("spm_hrf.R")
source("create-patch1.R")
source("create-patch2.R")
source("lc2num.R")
source("num2lc.R")
require("adlift")


effectuefastica<-function() {

par(mfrow=c(1,1))

require("fastICA")
source("fastICA.R")
source("get4ngb.R")

nblig<-15
nbcol<-15

# On cree les donnees
res<-brainsimul()

# matrice de taille TxM (T=nombre de temps)
donnees<-res$donnees
indices.patch1<-res$indices.patch1
indices.patch2<-res$indices.patch2
patch1.signal<-res$patch1.signal
patch2.signal<-res$patch2.signal


signal1<-res$signal1
signal2<-res$signal2




# On effectue l'ICA spatiale
resICAs<-fastICA(t(donnees),n.comp=2)

seuil<-qnorm(0.97)

par(mfrow=c(2,4))

X<-rep(1,nblig*nbcol)
X[indices.patch1]<-0
```

```
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)),main="Vrai patch 1")



# on utilise les z-score
X<-t(apply(matrix(scale(resICAs$S[,1]),nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2))
Y<-matrix(0,nrow=dim(X)[1],ncol=dim(X)[2])
Y[abs(X)>seuil]<-X[abs(X)>seuil]
image(Y,main="1ere composante de la SICA: z-scores")

image(t(apply(matrix(resICAs$S[,1],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="1ere composante de la SICA: image")


contour(t(apply(matrix(resICAs$S[,1],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="1ere composante de la SICA: contour")


X<-rep(1,nblig*nbcol)
X[indices.patch2]<-0
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)),main="Vrai patch 2")


# on utilise les z-score
X<-t(apply(matrix(scale(resICAs$S[,2]),nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2))
Y<-matrix(0,nrow=dim(X)[1],ncol=dim(X)[2])
Y[abs(X)>seuil]<-X[abs(X)>seuil]
image(Y,main="2eme composante de la SICA: z-scores")


image(t(apply(matrix(resICAs$S[,2],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="2eme composante de la SICA: image")


contour(t(apply(matrix(resICAs$S[,2],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="2eme composante de la SICA: contour")

# ------------------------------------------------------------------------------------------


X11()



# On effectue l'ICA temporelle
resICAt<-fastICA(donnees,n.comp=2)

par(mfrow=c(2,5))
```

```
plot(signal1,type="l",main="Vrai Signal patch1 (SP1)")


y<-resICAs$A[1,]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(y,type="l",main="Coefficients du premier axe principal\n de la SICA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-as.vector(denoise(1:length(signal1),resICAs$A[1,],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



y<-resICAt$S[,1]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(y,type="l",main="1ere composante de la TICA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-as.vector(denoise(1:length(signal1),resICAt$S[,1],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



plot(signal2,type="l",main="Vrai signal patch2 (SP2)")


y<-resICAs$A[2,]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(y,type="l",main="Coefficients du second axe principal\n de la SICA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))
```

```
y<-as.vector(denoise(1:length(signal1),resICAs$A[2,],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)



plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



y<-resICAt$S[,2]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(y,type="l",main="2eme composante de la TICA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))

y<-as.vector(denoise(1:length(signal1),resICAt$S[,2],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



}
```

F.2 Fonction lc2num

```
# lc2num : (ligne,colonne)'s vers numÃ©ro's
lc2num<-function(matrice,nblig,nbcol) {

# matrice est de taille 2*n
n<-ncol(matrice)
res<-rep(0,n)
for (i in 1:n) {
if (matrice[1,i] <= nblig & matrice[2,i] <= nbcol) res[i]<-(matrice[1,i]-1)*nbcol+matrice[2,i]
else stop("ligne ou colonne trop grand")
}

return(res)

}
```

F.3 Fonction plot.volume

```
require(AnalyzeFMRI)
require(marray)

# Permet d'afficher un volume cÃ©rÃ©bral anatomique ou fonctionnel
```

```
# ParamÃ¨tres d'entrÃ©e:
# file.or.mat: nom du fichier .img ou bien une array 3d contenant l'image volumique
# couleur: "anatomique" ou "fonctionnelle" suivant le type de l'image que l'on charge

plot.volume<-function(file.or.mat=
"../../MonCerveau/Lafaye/LAFAYE_7_1/LAFAYE_7_1-001.img"
,couleur="anatomique") {



if (couleur=="anatomique") col<-gray(seq(from=0.2,to=1,len=256))
if (couleur=="fonctionnelle") col<-heat.colors(256)



#if (couleur=="anatomique") colanat<-c(gray(seq(0,0.9,len=30)),seq(0,0,len=50))
#if (couleur=="fonctionnelle") colfonc<-c(gray(seq(0,0.9,len=30)))

if (is.character(file.or.mat)) {
my.brain<-f.read.analyze.volume(file.or.mat)
}
else {
if (length(dim(file.or.mat))==3) {my.brain<-array(file.or.mat,dim=c(dim(file.or.mat),1))}
else {my.brain<-file.or.mat}
}


mini<-min(my.brain)
maxi<-max(my.brain)


dimensions<-dim(my.brain)

coupe<-"sagittale"
nb.dim<-dimensions[1]


require("tkrplot")
tt <- tktoplevel()
nn<-1

img <-tkrplot(parent=tt, fun=function() {



nf <- layout(matrix(c(1, 2, 1, 0), 2, 2, byrow = TRUE),respect = F,widths=c(5,1))
par(mar=c(4,4,3,0))
par(mai=c(1, 1, 1, 0))

breaks<-seq(from=mini,to=maxi,len=length(col)+1)
```

```
if (coupe=="sagittale")
{image(1:dimensions[2],1:dimensions[3],my.brain[nn,,,1],col=col,breaks=breaks,xlab="",ylab="",asp=1)}
if (coupe=="coronale")
{image(1:dimensions[1],1:dimensions[3],my.brain[,nn,,1],col=col,breaks=breaks,xlab="",ylab="",asp=1)}
if (coupe=="axiale")
{image(1:dimensions[1],1:dimensions[2],my.brain[,,nn,1],col=col,breaks=breaks,xlab="",ylab="",asp=1)}


par(mai=c(0.8, 0.3, 0.1, 0.9))
maColorBar(x=seq(from=mini,to=maxi,len=length(col)+1),col=col,horizontal=F)



}
, hscale=1.7, vscale=1.7)






f<-function(...) {
         n <- as.numeric(tclvalue("nn"))
         if (n != nn) {
             nn <<- n
             tkrreplot(img)
         }
     }

SliderTo <- tclVar(nb.dim)


s <- tkscale(tt, command=f, from=1, to=as.numeric(tclvalue(SliderTo)), variable="nn",
                  showvalue=TRUE, resolution=1, orient="horiz")


OnOK <- function()
{
    tkdestroy(tt)
}
OK.but <- tkbutton(tt,text="Quitter",command=OnOK)

slicenb<-tklabel(tt,text="Slice number")

rb1 <- tkradiobutton(tt,command=function() {coupe<<-"axiale";if (nn>dimensions[3])
nn<<-dimensions[3];tkconfigure(s,to=dimensions[3]);tkrreplot(img)})
rb2 <- tkradiobutton(tt,command=function() {coupe<<-"coronale";if (nn>dimensions[2])
nn<<-dimensions[2];tkconfigure(s,to=dimensions[2]);tkrreplot(img)})
rb3 <- tkradiobutton(tt,command=function() {coupe<<-"sagittale";if (nn>dimensions[1])
nn<<-dimensions[1];tkconfigure(s,to=dimensions[1]);tkrreplot(img)})
```

```
rbValue <- tclVar("sagittale")
tkconfigure(rb1,variable=rbValue,value="axiale")
x1<-tklabel(tt,text="axiale ")
tkconfigure(rb2,variable=rbValue,value="coronale")
x2<-tklabel(tt,text="coronale ")
tkconfigure(rb3,variable=rbValue,value="sagittale")
x3<-tklabel(tt,text="sagittale ")


tkpack(img,slicenb,s,x3,rb3,x2,rb2,x1,rb1,OK.but)
tkpack.configure(x3,rb3,x2,rb2,x1,rb1, side="left")
tkpack.configure(OK.but, side="right")
}
```

F.4 Fonction brainsimul

```
brainsimul <- function(TR=1,block.length=24,n.blocks=5,sdnoise=1,nblig=15,nbcol=15,n=30,p=4,q=3) {


signal1<-signal.temporel(TR=TR,block.length=block.length,sdnoise=sdnoise,
signal.to.noise.ratio=0.9,type=1,plot=FALSE)
patch1.signal<-signal1$patch.signal
signal2<-signal.temporel(TR=TR,block.length=block.length,sdnoise=sdnoise,
signal.to.noise.ratio=0.9,type=2,plot=FALSE)
patch2.signal<-signal2$patch.signal


indices.patch1<-create.patch.1(nblig=nblig,nbcol=nbcol,p=p,q=q,plot=FALSE)
indices.patch2<-create.patch.2(n=n,nblig=nblig,nbcol=nbcol,plot=FALSE)

ts.length<-length(signal1$signal)

X<-as.list(1:ts.length)

res<-c()

for (i in 1:ts.length) {

X[[i]]<-rnorm(nblig*nbcol,sd=sdnoise) # bruit sur toute ma grille sur une seule ligne au temps 1


X[[i]][indices.patch1]<-X[[i]][indices.patch1]+patch1.signal[i]
X[[i]][indices.patch2]<-X[[i]][indices.patch2]+patch2.signal[i]

X[[i]]<-matrix(X[[i]],nrow=nblig,ncol=nbcol,byrow=T)

image(t(apply(X[[i]],FUN=rev,MARGIN=2)))

res<-rbind(res,as.vector(X[[i]]))
```

```
}


image(t(apply(X[[1]],FUN=rev,MARGIN=2)))


# renvoie une matrice ts.length x (nblig*nbcol)
return(list(donnees=res,indices.patch1=indices.patch1,indices.patch2=indices.patch2,
patch1.signal=patch1.signal,patch2.signal=patch2.signal,signal1=signal1$signal,signal2=signal2$signal))


}
```

F.5 Fonction fastICA

```
fastICA <- function (X, n.comp, alg.typ = c("parallel", "deflation"), fun = c("logcosh",
    "exp"), alpha = 1, method = c("R", "C"), row.norm = FALSE,
    maxit = 200, tol = 1e-04, verbose = FALSE, w.init = NULL)
{

gc(T)
    dd <- dim(X)
    d <- dd[dd != 1]
    if (length(d) != 2)
        stop("data must be matrix-conformal")
    X <- if (length(d) != length(dd))
        matrix(X, d[1], d[2])
    else as.matrix(X)
gc(T)
    if (alpha < 1 || alpha > 2)
        stop("alpha must be in range [1,2]")
    method <- match.arg(method)
    alg.typ <- match.arg(alg.typ)
    fun <- match.arg(fun)
    n <- nrow(X)
    p <- ncol(X)
    if (n.comp > min(n, p)) {
        cat("n.comp is too large\nn.comp set to", min(n, p),
            "\n")
        n.comp <- min(n, p)
    }
    if (is.null(w.init)) {
        w.init <- matrix(rnorm(n.comp^2), n.comp, n.comp)
    }
    else {
        if (!is.matrix(w.init) || length(w.init) != (n.comp^2))
            stop("w.init is not a matrix or is the wrong size")
    }
    if (method == "R") {
        if (verbose)
            cat("Centering\n")
        X <- scale(X, scale = FALSE)
```

```
 gc(T)
       if (row.norm) {
            X <- t(scale(X, scale = row.norm))
gc(T)
        }
        else {
            X <- t(X)
 gc(T)
       }

# Maintenant, X est de taille p x n

        if (verbose)
            cat("Whitening\n")
        V <- X %*% t(X)/n # de taille p x p
gc(T)
        s <- La.svd(V) # s$u et s$v de taille p x p, s$d de longueur p
gc(T)
        D <- diag(c(1/sqrt(s$d))) # de taille p x p ;
  #sqrt(n*s$d[1:(tm-1)])==svd(t(Xtcentree),nu=0,nv=0)$d[1:(tm-1)]
  # D == sqrt(n)*D.moins.1
gc(T)
        K <- D %*% t(s$u) # de taille n.comp x p
gc(T)
        K <- matrix(K[1:n.comp, ], n.comp, p) # de taille n.comp x p
gc(T)
        X1 <- K %*% X # de taille n.comp x n
 gc(T)
       if (alg.typ == "deflation") {
            a <- ica.R.def(X1, n.comp, tol = tol, fun = fun,
                alpha = alpha, maxit = maxit, verbose = verbose,
                w.init = w.init) # de taille n.comp x n.comp
gc(T)
        }
        else if (alg.typ == "parallel") {
            a <- ica.R.par(X1, n.comp, tol = tol, fun = fun,
                alpha = alpha, maxit = maxit, verbose = verbose,
                w.init = w.init) # de taille n.comp x n.comp
gc(T)
        }
        w <- a %*% K # de taille n.comp x p
gc(T)
        S <- w %*% X # de taille n.comp x n
gc(T)
        A <- t(w) %*% solve(w %*% t(w)) # de taille p x n.comp
gc(T)
        return(list(X = t(X), K = t(K), W = t(a), A = t(A), S = t(S)))
    }
    else if (method == "C") {
        a <- .C("icainc_JM", as.single(X), as.single(w.init),
            as.integer(p), as.integer(n), as.integer(n.comp),
```

```
            as.single(alpha), as.integer(1), as.integer(row.norm),
            as.integer(1 + (fun == "exp")), as.integer(maxit),
            as.single(tol), as.integer(alg.typ != "parallel"),
            as.integer(verbose), X = single(p * n), K = single(n.comp *
                p), W = single(n.comp * n.comp), A = single(p *
                n.comp), S = single(n.comp * n), PACKAGE = "fastICA")
gc(T)
        X1 <- t(matrix(a$X, p, n, byrow = TRUE))
gc(T)
        K <- t(matrix(a$K, n.comp, p, byrow = TRUE))
gc(T)
        W <- t(matrix(a$W, n.comp, n.comp, byrow = TRUE))
gc(T)
        A <- t(matrix(a$A, p, n.comp, byrow = TRUE))
gc(T)
        S <- t(matrix(a$S, n.comp, n, byrow = TRUE))
gc(T)
        return(list(X = X1, K = K, W = W, A = A, S = S))
    }
}


ica.R.par <- function (X, n.comp, tol, fun, alpha, maxit, verbose, w.init)
{
gc(T)
    n <- nrow(X)
    p <- ncol(X)
    W <- w.init # de taille n.comp x n.comp
    sW <- La.svd(W) # 3 matrices de taille n.comp x n.comp
    W <- sW$u %*% diag(1/sW$d) %*% t(sW$u) %*% W # de taille n.comp x n.comp
    W1 <- W # de taille n.comp x n.comp
    lim <- rep(1000, maxit)
    it <- 1
    if (fun == "logcosh") {
        if (verbose)
            cat("Symmetric FastICA using logcosh approx. to neg-entropy function\n")
        while (lim[it] > tol && it < maxit) {
            wx <- W %*% X # de taille n.comp x n
gc(T)
            gwx <- tanh(alpha * wx) # de taille n.comp x n
gc(T)
            v1 <- gwx %*% t(X)/p # de taille n.comp x n.comp
            g.wx <- alpha * (1 - (gwx)^2) # de taille n.comp x n
gc(T)
            v2 <- diag(apply(g.wx, 1, FUN = mean)) %*% W # de taille n.comp x n.comp
            W1 <- v1 - v2 # de taille n.comp x n.comp
            sW1 <- La.svd(W1)
            W1 <- sW1$u %*% diag(1/sW1$d) %*% t(sW1$u) %*% W1 # de taille n.comp x n.comp
            lim[it + 1] <- max(Mod(Mod(diag(W1 %*% t(W))) - 1))
            W <- W1
            if (verbose)
```

```
                cat("Iteration", it, "tol =", format(lim[it +
                  1]), "\n")
            it <- it + 1
        }
    }
    if (fun == "exp") {
        if (verbose)
            cat("Symmetric FastICA using exponential approx. to neg-entropy function\n")
        while (lim[it] > tol && it < maxit) {
            wx <- W %*% X
            gwx <- wx * exp(-(wx^2)/2)
            v1 <- gwx %*% t(X)/p
            g.wx <- (1 - wx^2) * exp(-(wx^2)/2)
            v2 <- diag(apply(g.wx, 1, FUN = mean)) %*% W
            W1 <- v1 - v2
            sW1 <- La.svd(W1)
            W1 <- sW1$u %*% diag(1/sW1$d) %*% t(sW1$u) %*% W1
            lim[it + 1] <- max(Mod(Mod(diag(W1 %*% t(W))) - 1))
            W <- W1
            if (verbose)
                cat("Iteration", it, "tol =", format(lim[it +
                  1]), "\n")
            it <- it + 1
        }
    }
    return(W)
}
```

F.6 Fonction signal.temporel

```
signal.temporel <- function(TR=2,block.length=20,n.blocks=4,sdnoise=1,
signal.to.noise.ratio=0.9,type=1,plot=FALSE) {

# Exemple:
# TR=2  # repetition time, in seconds (temps entre le debut d'aquisition d'une image
# et le debut d'aquisition de l'image suivante)
# block.length=6 # block length, in seconds
# Donc, il y a eu block.length/TR=3 images acquises dans chaque bloc
# n.blocks=4  # number of task blocks (uniquement le nombre de blocs de la tache!)

# create boxcar to describe the blocked design
block.length=round(block.length/TR)  # change time units to TRs
n.scans=n.blocks*block.length*2 + block.length



# Signal a tendance quadratique
if (type==1) {
signal<-(1:n.scans-n.scans/2)**2
}

# Signal crÃ©neau
```

```
if (type==2) {
boxcar=rep(0,n.scans)
for (b in 1:n.blocks) {
  boxcar[(b*block.length*2+1 - block.length):(b*block.length*2)]=1
}
signal<-boxcar
}


# Signal convolue avec une reponse hemodynamique
hrf.data<-spm.hrf(TR)$hrf  # get the HRF from spm


conv.signal.hrf<-rev(convolve(rev(signal),hrf.data,type="open"))  # do the convolution
signal.convolue<-conv.signal.hrf[1:length(signal)]   # trim off the extra points added by convolve()

patch.signal<-signal.convolue/(sd(signal.convolue))*sdnoise*signal.to.noise.ratio


if (plot) {
par(mfrow=c(3,1))
plot(seq(from=0,by=TR,length=length(signal)),signal,type="b",col="blue",ylim=c(min(signal.convolue),
max(signal.convolue)),xlim=c(0,30))
plot(seq(from=0,by=TR,length=length(hrf.data)),hrf.data,type="b")
plot(seq(from=0,by=TR,length=length(signal)),signal,type="b",col="blue",ylim=c(min(signal.convolue),
max(signal.convolue)),xlab="Time (en TR/sec)")
title("Signal et sa convolution par la HRF")
points(seq(from=0,by=TR,length=length(signal)),signal.convolue,type="b",col="red")
}



return(list(patch.signal=patch.signal,signal=signal,signal.convolue=signal.convolue))


}
```

F.7 Fonction create.patch.1

```
create.patch.1 <- function(nblig=15,nbcol=15,p=4,q=3,plot=FALSE) {

if (p>nblig) stop("You should have p<=nblig")
if (q>nbcol) stop("You should have q<=ncol")

# vecteur (r,c)=(ligne,colonne) de localisation du coin supÃ©rieur gauche du patch rectangulaire.
# r<=(nblig-p+1) et c<=(nbcol-q+1)
squaretopleft<-c(round(runif(1)*(nblig-p))+1,round(runif(1)*(nbcol-q))+1)
```

```
indices.patch1<-as.vector(apply(cbind(as.matrix(lc2num(as.matrix(squaretopleft),
nblig,nbcol)+nbcol*0:(p-1)),matrix(1,nrow=p,ncol=(q-1))),FUN=cumsum,MARGIN=1))

if (plot) {
X<-rep(0,nblig*nbcol)
X[indices.patch1]<-1
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)))
}

return(indices.patch1)

}
```

F.8 Fonction foncsuranat2

```
# Calcule la matrice dont les voxels contiennent les z-scores Ã  partir de la matrice 3D fonctionnelle
# de la composante donnee en entree (ICA ou ACP).

# Calcule les coordonnees des voxels "actifs" de la matrice fonctionnelle 3D dont la valeur est
# superieure Ã  un seuil.

# Met la valeur de chaque voxel actif de la matrice fonctionnelle 3D dans le pool de voxels
# correspondants de la matrice anatomique 3D.

# Renvoie cette nouvelle matrice 3D anatomo-fonctionnelle.



recup.indices<-function(X){c(slice.index(X,1),slice.index(X,2),slice.index(X,3))}


foncsuranat2 <- function(volume.fonc,volume.anat,Mfonc,Manat,seuil) {

# volume.fonc: array contenant l'image volumique fonctionnelle
# volume.anat: array contenant l'image volumique anatomique
# Mfonc: matrice 4x4 du fichier .mat associe Ã  la fonctionnelle
# Manat: matrice 4x4 du fichier .mat associe Ã  l'anatomique
# seuil: valeur du seuil au dessus duquel on considÃ¨re les voxels comm etant actifs



if (length(dim(volume.fonc))==4 & dim(volume.fonc)[4]==1) volume.fonc<-volume.fonc[,,,1]
if (length(dim(volume.anat))==4 & dim(volume.anat)[4]==1) volume.anat<-volume.anat[,,,1]


res<-array(NA,dim=dim(volume.anat))
```

```
res[cbind(matrix(recup.indices(volume.fonc)[volume.fonc>seuil],ncol=3),1)%*%
t(Mfonc)%*%t(solve(Manat)[1:3,])]<-volume.fonc[volume.fonc>seuil]*100000


return(res)

}
```

F.9 Fonction liredata

```
require(AnalyzeFMRI)

# Permet de lire les T=70 images 3D du cerveau contenant M=64*64*32 voxels
# et les renvoie dans une matrice de taille T*M
# Les images sont contenues dans les fichiers nom*

liredata<-function(path="../../MonCerveau/Lafaye/LAFAYE_3_1/",prefix="LAFAYE_3_1-",
regexp="???.img",Temps=1:70,dim="4d") {


noms.fichiers<-list.files(path=path,pattern=glob2rx(paste(prefix,regexp,sep="")))[Temps]

my.brain<-c()
for (nom in noms.fichiers) {
if (dim=="3d") my.brain<-cbind(my.brain,matrix(f.read.analyze.volume(paste(path,nom,sep=""))[,,]))
if (dim=="4d") my.brain<-cbind(my.brain,matrix(f.read.analyze.volume(paste(path,nom,sep=""))[,,,1]))
}


return(my.brain)

}
```

F.10 Fonction spm.hrf

```
spm.hrf <- function(RT) {
# returns a hemodynamic response function
# RT   - scan repeat time
# p    - parameters of the response function (two gamma functions)
#
# defaults
# (seconds)
# p(1) - delay of response (relative to onset)    6
# p(2) - delay of undershoot (relative to onset)    16
# p(3) - dispersion of response    1
# p(4) - dispersion of undershoot    1
# p(5) - ratio of response to undershoot    6
# p(6) - onset (seconds)    0
# p(7) - length of kernel (seconds)   32
#
```

```
# hrf  - hemodynamic response function
# p    - parameters of the response function



fMRI.T = 16 # number of time-bins per scan used when building regressors

p     = c(6,16,1,1,6,0,32)

# modelled hemodynamic response function - {mixture of Gammas}
#-------------------------------------------------------------------------
dt    = RT/fMRI.T
u     = 0:(p[7]/dt) - p[6]/dt
hrf   = dgamma(u,p[1]/p[3],dt/p[3]) - dgamma(u,p[2]/p[4],dt/p[4])/p[5]
hrf   = hrf[(0:(p[7]/RT))*fMRI.T + 1]
hrf   = hrf/sum(hrf) # on la normalise POURQUOI??? IL FAUT FAIRE PAREIL DANS LE PARADIGME??


return(list(hrf=hrf,p=p))

}
```

F.11 Fonction create.patch.2

```
# Fonction pour generer les indices des cases du motif 2
create.patch.2 <- function(n=30,nblig=15,nbcol=15,plot=FALSE) {

if (n>=nblig*nbcol) stop("n est trop grand")

#step1: case1=(r1,c1) tire uniformement sur la grille 15x15
#step2: case2=(r2,c2) tire uniformement sur la grille 15x15, parmi les voisins de case1
#step3: case3=(r3,c3) tire uniformement sur la grille 15x15, parmi les voisins de {case1,case2}
#step4: case4=(r4,c4) tire uniformement sur la grille 15x15, parmi
# les voisins de {case1,case2,case3}
#etc ...

#Note:
#les voisins de {case1,case2} = les voisins de case1 union
# les voisins de case2 prive de {case1,case2}
#les voisins de {case1,case2,case3} = les voisins de {case1,case2} union
# les voisins de case3 prive de {case1,case2,case3}
#etc ...


cases<-matrix(0,nrow=2,ncol=n,byrow=F)
indices.patch2<-rep(0,n)

indices.patch2[1]<-round(runif(1)*(nblig*nbcol-1))+1
cases[,1]<-num2lc(indices.patch2[1],nblig,nbcol) # on tire au hasard la premiÃ¨re case
voisinscases<-get4ngb(nblig,nbcol,cases[1,1],cases[2,1])
```

```
indicesvoisins<-lc2num(as.matrix(voisinscases),nblig,nbcol)

voisinscases<-num2lc("%w/o%"(indicesvoisins,indices.patch2),nblig,nbcol)



for (j in 2:n) {



cases[,j]<-voisinscases[,round(runif(1)*(ncol(voisinscases)-1))+1]

indices.patch2[j]<-lc2num(as.matrix(cases[,j]),nblig,nbcol)

voisinscases<-cbind(voisinscases,get4ngb(nblig,nbcol,cases[1,j],cases[2,j]))


indicesvoisins<-lc2num(as.matrix(voisinscases),nblig,nbcol)

voisinscases<-num2lc("%w/o%"(indicesvoisins,indices.patch2),nblig,nbcol)



}


if (plot) {
X<-rep(0,nblig*nbcol)
X[indices.patch2]<-1
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)))
}

return(indices.patch2)

}


"%w/o%" <- function(x,y) x[!x %in% y] #--  x without y
```

F.12 Fonction get4ngb

```
get4ngb <- function(rows,cols,x,y) {

  # function ngb = voisins(rows,cols,x,y)
  # This function returns the coordinates of the 4-neighbours
  # of an element in a matrix.  The values are returned in the
  # list ngb.  If the neighbour does not exist,- that is (x,y)
  # corredsponds to an edge or corner of the array,
  # the list of neighbours contains only those coordinates corresponding
  # to real neighbours.
```

```
  # Copyright Lars Aurdal/FFIE.

  # Handle left edge.

  # Handle upper left corner.

ngb = matrix(0,nrow=2,ncol=4)

  if ((x ==  1) & (y == 1)) {
    ngb[1:2,1:1] = c(2,1)
    ngb[1:2,2:2] = c(1,2)}

    # Handle lower left corner.

  else if ((x == rows) & (y == 1)) {
    ngb[1:2,1:1] = c(rows,2)
    ngb[1:2,2:2] = c((rows - 1),1)}

    # Handle left edge in general.

  else if (y == 1) {
    ngb[1:2,1:1] = c((x+1),1)
    ngb[1:2,2:2] = c(x,2)
    ngb[1:2,3:3] = c((x-1),1)}

    # Handle right edge.

    # Handle upper right corner.

  else if ((x == 1) & (y == cols)) {
    ngb[1:2,1:1] = c(1,(cols-1))
    ngb[1:2,2:2] = c(2,cols)}

    # Handle lower right corner.

  else if ((x == rows) & (y == cols)) {
    ngb[1:2,1:1] = c(rows,(cols-1))
    ngb[1:2,2:2] = c((rows-1),cols)}

    # Handle right edge in general.

  else if (y == cols) {
    ngb[1:2,1:1] = c((x+1),cols)
    ngb[1:2,2:2] = c(x,(cols-1))
    ngb[1:2,3:3] = c((x-1),cols)}

    # Handle top line.

  else if (x == 1) {
    ngb[1:2,1:1] = c(1,(y-1))
    ngb[1:2,2:2] = c(2,y)
```

```
    ngb[1:2,3:3] = c(1,(y+1))}

    # Handle bottom line.

  else if (x == rows) {
    ngb[1:2,1:1] = c(rows,(y-1))
    ngb[1:2,2:2] = c((rows-1),y)
    ngb[1:2,3:3] = c(rows,(y+1))}

    # Handle general case

  else {
    ngb[1:2,1:1] = c((x-1),y)
    ngb[1:2,2:2] = c(x,(y-1))
    ngb[1:2,3:3] = c((x+1),y)
    ngb[1:2,4:4] = c(x,(y+1))}


voisins<-ngb[,ngb[1,]!=0]


  return(voisins)

}
```

F.13 Fonction num2lc

```
# num2lc: numero's vers (ligne,colonne)'s
num2lc<-function(numero,nblig,nbcol) {

n<-length(numero)
res<-matrix(0,nrow=2,ncol=n)
for (i in 1:n) {
if (numero[i] <= nblig*nbcol) res[,i]<-c(numero[i]%/%nbcol+1,numero[i]%%nbcol)
else stop("numero trop grand")
if (res[2,i]==0) res[,i]<-c(res[1,i]-1,nbcol)
}

# res est de taille 2*n
return(res)

}
```

F.14 Fonction effectueacp

```
effectueacp<-function() {

par(mfrow=c(1,1))

source("acpxqd.r")
source("dcentred.r")
source("get4ngb.R")
```

```
require("adlift")


nblig<-15
nbcol<-15

# On cree les donnees
res<-brainsimul()

donnees<-res$donnees
indices.patch1<-res$indices.patch1
indices.patch2<-res$indices.patch2
patch1.signal<-res$patch1.signal
patch2.signal<-res$patch2.signal

signal1<-res$signal1
signal2<-res$signal2




# On effectue l'ACP spatiale
resPCAs<-acpxqd(t(donnees),impres=F,graph=F)




seuil<-qnorm(0.97)



par(mfrow=c(2,4))

X<-rep(NA,nblig*nbcol)
X[indices.patch1]<-1
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)),main="Vrai patch 1")



# on utilise les z-score
X<-t(apply(matrix(scale(resPCAs[[11]][,1]),nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2))
Y<-matrix(0,nrow=dim(X)[1],ncol=dim(X)[2])
Y[abs(X)>seuil]<-X[abs(X)>seuil]
image(Y,main="1ere composante de la SPCA: z-scores")


image(t(apply(matrix(resPCAs[[11]][,1],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="1ere composante de la SPCA: image")
```

```
contour(t(apply(matrix(resPCAs[[11]][,1],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="1ere composante de la SPCA: contour")



X<-rep(NA,nblig*nbcol)
X[indices.patch2]<-1
X<-matrix(X,nrow=nblig,ncol=nbcol,byrow=T)
image(t(apply(X,FUN=rev,MARGIN=2)),main="Vrai patch 2")


# on utilise les z-score
X<-t(apply(matrix(scale(resPCAs[[11]][,2]),nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2))
Y<-matrix(0,nrow=dim(X)[1],ncol=dim(X)[2])
Y[abs(X)>seuil]<-X[abs(X)>seuil]
image(Y,main="2eme composante de la SPCA: z-scores")



image(t(apply(matrix(resPCAs[[11]][,2],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="2eme composante de la SPCA: image")


contour(t(apply(matrix(resPCAs[[11]][,2],nrow=15,ncol=15,byrow=F),FUN=rev,MARGIN=2)),
main="2eme composante de la SPCA: contour")


# ------------------------------------------------------------------------------------------


X11()

# On effectue l'ACP temporelle
resPCAt<-acpxqd(donnees,impres=F,graph=F)


par(mfrow=c(2,5))

plot(signal1,type="l",main="Vrai Signal patch1 (SP1)")

y<-resPCAs[[10]][,1]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(y,type="l",main="Coefficients du premier axe principal\n de la SPCA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-as.vector(denoise(1:length(signal1),resPCAs[[10]][,1],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
```

```
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-resPCAt[[11]][,1]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(y,type="l",main="1ere composante de la TPCA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-as.vector(denoise(1:length(signal1),resPCAt[[11]][,1],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)

plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



plot(signal2,type="l",main="Vrai signal patch2 (SP2)")


y<-resPCAs[[10]][,2]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(y,type="l",main="Coefficients du second axe principal\n de la SPCA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))


y<-as.vector(denoise(1:length(signal1),resPCAs[[10]][,2],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)



plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))



y<-resPCAt[[11]][,2]
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)
```

```
plot(y,type="l",main="2eme composante de la TPCA",ylab="",
xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))

y<-as.vector(denoise(1:length(signal1),resPCAt[[11]][,2],AdaptNeigh,1,TRUE,TRUE,2)$fhat$coeff)
corrsp1<-round(max(abs(ccf(y,signal1,plot=FALSE)[[1]])),3)
corrsp2<-round(max(abs(ccf(y,signal2,plot=FALSE)[[1]])),3)


plot(1:length(signal1),y,type="l",main="Debruitage du signal de gauche\n par ondelettes",
ylab="",xlab=paste("Corr avec SP1: ",corrsp1," , avec SP2: ",corrsp2,sep=""))




}
```

F.15 Fonction hrf

```
peak<-function(t,mu1,sigma1) {

exp(-((t-mu1)/sigma1)^2/2)/(sigma1*sqrt(2*pi))

}


undershoot <- function(t,mu2,sigma2) {

exp(-((t-mu2)/sigma2)^2/2)/(sigma2*sqrt(2*pi))

}

hrf <- function(t,mu1,sigma1,mu2,sigma2,v,w) {


v*peak(t,mu1,sigma1)-w*undershoot(t,mu2,sigma2)

}


#x<-seq(from=0,to=17,length=17);
#hrf.data<-hrf(x,4,0.9,9,2.5,1.2,0.4)
#plot(hrf.data,type="l",xlim=c(0,18),col="blue")


#boxcar<-rep(c(rep(0,12),rep(1,12)),10)
#plot(boxcar,type="l",ylim=c(-0.5,1.5),col="blue")

#conv.boxcar.hrf<-convolve(boxcar,hrf.data,type="filter")
#points(conv.boxcar.hrf,type="l",col="red")
```

F.16 Fonction plot.volumefoncsuranat

```
require(AnalyzeFMRI)
require(marray)


# Permet d'afficher l'image volumique fonctionnelle superposee Ã  l'image volumique anatomique
# ParamÃ¨tres d'entree:
# volume.anat: array 3d de l'image anatomique
# volumeNAanatfonc: array 3d de l'image fonctionnelle qui ne contient que des NA
# sauf aux voxels actives
# ou il y a les valeurs d'activation
#  cette array est fabriquee a l'aide du programme foncsuranat.R

plot.volumefoncsuranat<-function(volume.anat,volumeNAanatfonc) {

if (length(dim(volume.anat))==4 & dim(volume.anat)[4]==1) volume.anat<-volume.anat[,,,1]
if (length(dim(volumeNAanatfonc))==4 & dim(volumeNAanatfonc)[4]==1)
{volumeNAanatfonc<-volumeNAanatfonc[,,,1]}



colfoncpos<-heat.colors(128)
colfoncneg<-rgb(0.6, 1:128/128,1)



colanat<-gray(seq(from=0.2,to=1,len=256))



minianat<-min(volume.anat,na.rm=T)
maxianat<-max(volume.anat,na.rm=T)

minifoncpos<-min(volumeNAanatfonc[volumeNAanatfonc>=0],na.rm=T)
maxifoncpos<-max(volumeNAanatfonc[volumeNAanatfonc>=0],na.rm=T)

minifoncneg<-min(volumeNAanatfonc[volumeNAanatfonc<0],na.rm=T)
maxifoncneg<-max(volumeNAanatfonc[volumeNAanatfonc<0],na.rm=T)


dimensions<-dim(volume.anat)

coupe<-"sagittale"
nb.dim<-dimensions[1]


require("tkrplot")
tt <- tktoplevel()
nn<-1

img <-tkrplot(parent=tt, fun=function() {
```

```
nf <- layout(matrix(c(1, 2, 1, 3, 1, 4), 3, 2, byrow = TRUE),respect = F,widths=c(5,1))
par(mar=c(4,4,3,0))
par(mai=c(1, 1, 1, 0))

breaksanat<-seq(from=minianat,to=maxianat,len=length(colanat)+1)
breaksfoncpos<-seq(from=minifoncpos,to=maxifoncpos,len=length(colfoncpos)+1)
breaksfoncneg<-seq(from=minifoncneg,to=maxifoncneg,len=length(colfoncneg)+1)

breaksfonc<-c(breaksfoncneg[-length(breaksfoncneg)],breaksfoncpos)
colfonc<-c(colfoncneg,colfoncpos)

if (minifoncneg==Inf) {
breaksfonc<-breaksfoncpos
colfonc<-colfoncpos
}

if (minifoncpos==-Inf) {
breaksfonc<-breaksfoncneg
colfonc<-colfoncneg
}




if (coupe=="sagittale") {image(1:dimensions[2],1:dimensions[3],volume.anat[nn,,],col=colanat,
breaks=breaksanat,xlab="",ylab="",asp=1)}

if (coupe=="coronale") {image(1:dimensions[1],1:dimensions[3],volume.anat[,nn,],col=colanat,
breaks=breaksanat,xlab="",ylab="",asp=1)}

if (coupe=="axiale") {image(1:dimensions[1],1:dimensions[2],volume.anat[,,nn],col=colanat,
breaks=breaksanat,xlab="",ylab="",asp=1)}



if (minifoncneg!=Inf | minifoncpos!=-Inf) {

if (coupe=="sagittale") {image(1:dimensions[2],1:dimensions[3],volumeNAanatfonc[nn,,],
col=colfonc,breaks=breaksfonc,xlab="",ylab="",asp=1,add=T)}

if (coupe=="coronale")  {image(1:dimensions[1],1:dimensions[3],volumeNAanatfonc[,nn,],
col=colfonc,breaks=breaksfonc,xlab="",ylab="",asp=1,add=T)}

if (coupe=="axiale") {image(1:dimensions[1],1:dimensions[2],volumeNAanatfonc[,,nn],
col=colfonc,breaks=breaksfonc,xlab="",ylab="",asp=1,add=T)}
```

```
}



par(mai=c(0.8, 0.3, 0.1, 0.9))
maColorBar(x=seq(from=minianat,to=maxianat,len=length(colanat)+1),col=colanat,horizontal=F)

if (minifoncpos!=-Inf) {
par(mai=c(0.8, 0.3, 0.1, 0.9))
maColorBar(x=seq(from=minifoncpos,to=maxifoncpos,len=length(colfoncpos)+1),col=colfoncpos,horizontal=F)
}

if (minifoncneg!=Inf) {
par(mai=c(0.8, 0.3, 0.1, 0.9))
maColorBar(x=seq(from=minifoncneg,to=maxifoncneg,len=length(colfoncneg)+1),col=colfoncneg,horizontal=F)
}

}
, hscale=1.7, vscale=1.7)






f<-function(...) {
         n <- as.numeric(tclvalue("nn"))
         if (n != nn) {
             nn <<- n
             tkrreplot(img)
         }
     }

SliderTo <- tclVar(nb.dim)


s <- tkscale(tt, command=f, from=1, to=as.numeric(tclvalue(SliderTo)), variable="nn",
                  showvalue=TRUE, resolution=1, orient="horiz")


OnOK <- function()
{
    tkdestroy(tt)
}
OK.but <- tkbutton(tt,text="Quitter",command=OnOK)

slicenb<-tklabel(tt,text="Slice number")

rb1 <- tkradiobutton(tt,command=function() {coupe<<-"axiale";if (nn>dimensions[3])
nn<<-dimensions[3];tkconfigure(s,to=dimensions[3]);tkrreplot(img)})
rb2 <- tkradiobutton(tt,command=function() {coupe<<-"coronale";if (nn>dimensions[2])
```

```
nn<<-dimensions[2];tkconfigure(s,to=dimensions[2]);tkrreplot(img)})
rb3 <- tkradiobutton(tt,command=function() {coupe<<-"sagittale";if (nn>dimensions[1])
nn<<-dimensions[1];tkconfigure(s,to=dimensions[1]);tkrreplot(img)})

rbValue <- tclVar("sagittale")
tkconfigure(rb1,variable=rbValue,value="axiale")
x1<-tklabel(tt,text="axiale ")
tkconfigure(rb2,variable=rbValue,value="coronale")
x2<-tklabel(tt,text="coronale ")
tkconfigure(rb3,variable=rbValue,value="sagittale")
x3<-tklabel(tt,text="sagittale ")


tkpack(img,slicenb,s,x3,rb3,x2,rb2,x1,rb1,OK.but)
tkpack.configure(x3,rb3,x2,rb2,x1,rb1, side="left")
tkpack.configure(OK.but, side="right")
}
```