

# TP de Traitement d'images avancé - IRM fonctionnelle

## Introduction

Nous allons durant ce TP mettre en oeuvre les notions de modèle linéaire général, de contraste et de carte statistique vues en cours. Pour cela nous utiliserons une acquisition IRMf réalisée par Geraint Rees sous la direction de Karl Friston (Functional Imaging Laboratory). Le sujet de leur étude était d'explorer les possibilités offertes par l'imagerie IRM fonctionnelle à ses débuts. Les données sont mises à disposition du monde universitaire sur le site <http://www.fil.ion.ucl.ac.uk/spm/data/auditory/>. Le jeu de données fourni a été acquis avec un IRM 2T Siemens. Chaque acquisition (3D) est de taille  $64 \times 64 \times 64$  avec des voxels de taille  $3 \times 3 \times 3 \text{ mm}^3$ . Chaque acquisition 3D prend en réalité 6.05s (pour les 64 coupes), les acquisitions ont finalement été réalisées avec un temps de répétition  $TR = 7\text{s}$ . Dans le jeu de données étudié, il y a 96 acquisitions réalisées sur un sujet avec un paradigme par blocs. Chaque bloc est composé de 6 acquisitions de silence puis 6 acquisitions durant lesquels le patient entend des mots bisyllabiques. Cela donne un paradigme de 8 blocs de 84s.

Dans ce TP, les données ont été prétraitées au préalable, le premier bloc de 12 acquisitions a été supprimé et les images ont été rognées sur les bords (des cubes 3D).

L'objectif de ce TP est d'étudier les zones cérébrales activées par l'audition de mots par le sujet dans l'IRM.

## Quelques fonctions Matlab utiles

Vous êtes invités à consulter régulièrement la documentation des différentes fonctions Matlab pour comprendre les opérations qu'elles effectuent :

```
>> help function    % Pour une description succincte
>> doc function    % Pour une description plus détaillée et des exemples
```

On rappelle quelques fonctions Matlab utiles à la manipulation des images IRMf dans ce TP.

```
>> load('image.mat', 'I');      # Chargement de l'image I contenue dans le
fichier 'image.mat'.

>> [N1, N2, N3, N4] = size(I);  # Calcul de la taille de l'image en 4D et
stockage dans les variables N1, N2, N3, N4.

>> I(i,:,:,t0);                # Sélection de la coupe sagittale d'indice
i de l'image I à l'instant t0.

>> I(:,i,:,:,t0);              # Sélection de la coupe coronale d'indice
i de l'image I à l'instant t0.

>> I(:,:,:,i,t0);              # Sélection de la coupe axiale d'indice i
de l'image I à l'instant t0.
```

```
>> I(i1:i2, j1:j2, i, t0);      # Sélection de la partie de la coupe axiale
d'indice i à l'instant t0 comprise entre les lignes i1 et i2 et les colonne j1
et j2.

>> imshow(I, [ ]);            # Affichage de l'image I, ne pas oublier
l'option [ ] pour adapter l'échelle des niveaux de gris à la dynamique de
l'image.

>> imshowpair(I1, I2);        # Superposition de l'image I2 sur l'image
I1.

>> subplot(P,Q, i); imshow(...); # Affichage de plusieurs images sur P
lignes et Q colonnes. L'indice i varie de 1 à  $P \times Q$ .

>> c = conv(s, h, 'same');     # Convolue le signal s avec le signal h et
renvoie le résultat dans un vecteur c de même taille que le signal s.

>> X' ;                       # Transposée de la matrice X.
```

## 1 Préparation des données

Les données prétraitées sont disponibles dans l'archive 'data.zip'. Décompresser l'archive et placer les fichiers dans votre répertoire de travail courant.

### 1.1 Chargement des images et signaux utiles dans ce TP

#### Manipulation 1

---

- ➔ Charger l'acquisition IRMf contenue dans le fichier 'swfM00223.mat', la matrice 4D est stockée dans la variable 'I'.
  - ➔ Enregistrer les dimensions de l'image I dans 4 variables variables N1, N2, N3, N4.
  - ➔ Charger le masque contenu dans le fichier 'mask.mat', le masque 3D est contenu dans la variable 'mask'.
  - ➔ Charger la réponse hémodynamique contenue dans le fichier 'hrf.mat', le vecteur est stocké dans la variable 'hrf'.
- 

### 1.2 Réponse hémodynamique

#### Manipulation 2

---

- ➔ Quelle est la fréquence d'échantillonnage temporel des données ?
  - ➔ Afficher la réponse hémodynamique fournie (elle a été échantillonnée avec la même fréquence que les données). Que peut-on dire sur cette réponse et sur la fréquence d'échantillonnage temporel de l'acquisition IRMf ?
-

### 1.3 Extraction des décours temporels du cerveau

Avant d'analyser l'image IRMf, nous devons extraire les voxels appartenant au cerveau (la boîte crânienne et tout ce qui se trouve à la périphérie du cerveau constituent une nuisance pour l'analyse). On fournit un masque binaire dans le fichier 'mask.mat' qui contient des 1 dans les voxels appartenant au cerveau et des 0 partout ailleurs.

---

**Manipulation 3**

- ➔ Combien de voxels contient le masque? On stockera cette valeur dans la variable `nbVox`.
  - ➔ Créer une matrice `Y` de taille  $N4 \times nbVox$  qui contiendra dans chaque colonne les décours temporels des voxels du cerveau.
- 

Afin d'extraire les décours temporels d'intérêt on utilisera le code suivant :

```
>> for i = 1:N4
    tmp = squeeze(I(:,:,i)); #Extraction de l'acquisition 3D au temps i.
    Y(i,:) = tmp(mask > 0);
end
```

---

**Manipulation 4**

- ➔ Extraire les décours temporels avec le code ci-dessus.
- 

On travaillera désormais avec cette matrice `Y` où chaque colonne correspond à un décours temporel de taille  $N4 \times 1$  (vecteur  $y$  dans les formules du cours).

## 2 Modèle linéaire généralisé

### 2.1 Création du dessin du paradigme

D'après la description du paradigme expérimental, on veut créer un vecteur qui reflète l'alternance de séquences audio et des séquences de silence.

---

**Manipulation 5**

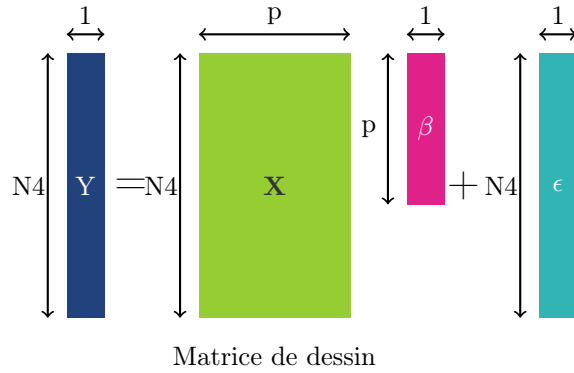
- ➔ Créer le vecteur `paradigme` contenant le paradigme expérimental décrit en introduction.
  - ➔ Afficher ce vecteur et vérifier qu'il est conforme à l'expérience (la régression est d'autant meilleure que le vecteur modélisant le paradigme est précis).
  - ➔ Ce vecteur `paradigme` peut être convolué à la réponse hémodynamique afin de synthétiser au mieux la réponse attendue à la stimulation audio. On appellera `paradigme_hrf` le paradigme convolué à la réponse hémodynamique contenue dans le vecteur `hrf`.
- 

### 2.2 Création de la matrice de dessin

On rappelle que le modèle linéaire généralisé pour un unique décours temporel s'écrit :

$$Y = \sum_i \beta_i x_i + \epsilon = X\beta + \epsilon \quad (1)$$

où les  $x_i$  sont des régresseurs connus (choisis en fonction du paradigme!),  $\beta_i$  sont les paramètres inconnus à estimer. On peut représenter le modèle (1) graphiquement :



Dans la matrice  $X$  on souhaite utiliser  $p = 3$  régresseurs :

- un vecteur constant (pour modéliser un éventuel offset, ou la moyenne du bruit si celui-ci est non nul, etc),
- le vecteur de paradigme (convolué ou non à la réponse hémodynamique)
- un vecteur **meanY** qui contient la moyenne des décours temporels.

Le dernier régresseur est lié à la présence d'un hypersignal dans les ventricules (remplis de liquide céphalo-rachidien) qui se retrouve dans la moyenne des décours temporels.

### Manipulation 6

➔ Construire les 3 régresseurs sous la forme de 3 vecteurs  $x_1, x_2$  et  $x_3$  de taille  $N4 \times 1$  contenant les informations données ci-dessus.

➔ Construire la matrice  $X$  en mettant dans la première colonne le vecteur  $x_1$ , dans la deuxième colonne le vecteur  $x_2$  et dans la troisième colonne, le vecteur  $x_3$ .

## 2.3 Estimation au sens des moindres carrés

L'estimateur au sens des moindres carrés du vecteur  $\beta$  contenant les 3 coefficients de régression est obtenu par l'équation :

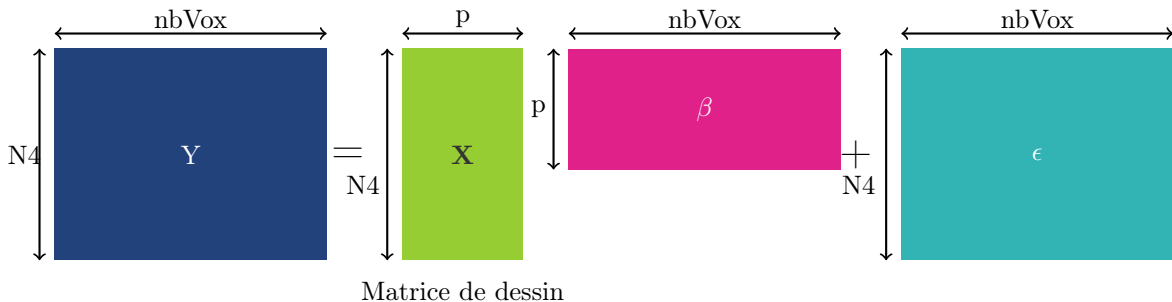
$$\hat{\beta} = (X^T X)^{-1} X^T y$$

lorsque  $y$  contient un unique décours temporel.

On souhaite calculer ici les 3 coefficients de régression pour l'ensemble des voxels du cerveau contenus dans la matrice  $Y$  de taille  $N4 \times nbVox$ . Le calcul de  $\hat{\beta}$  peut se faire pour l'ensemble des voxels en réécrivant le problème sous forme matricielle :

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{2}$$

Le modèle (2) peut se représenter graphiquement :



### Manipulation 7

- ➔ Calculer la matrice  $\hat{\beta}$ .
  - ➔ Comparer la taille de la matrice  $\hat{\beta}$  à la taille du masque 3D du cerveau.
  - ➔ Conclure sur la façon de visualiser en 3D les valeurs de  $\hat{\beta}$  pour chaque régresseur.
- 

## 2.4 Carte de contraste

On s'intéresse au contraste  $\beta_2 - \beta_3$  où  $\beta_2$  est le coefficient de régression lié au paradigme et  $\beta_3$  est le coefficient de régression lié à la valeur moyenne des décours temporels.

### Manipulation 8

- ➔ Créer le vecteur  $c$  de contraste.
- 

On forme la statistique de décision :

$$T = \frac{c^T \hat{\beta}}{\sqrt{\hat{\sigma}^2 c^T (X^T X)^{-1} c}}$$

avec

$$\hat{\sigma}^2 = \frac{1}{n - p - 1} \|Y - X\hat{\beta}\|_2^2$$

Dans le cas matriciel (modèle (2)),  $T$  et  $\hat{\sigma}^2$  sont des vecteurs de taille  $nbVox \times 1$ .

La loi  $p(T|\mathcal{H}_0)$  de la statistique  $T$  sous  $\mathcal{H}_0$  est la loi de Student à  $(N4 - p - 2)$  degrés de liberté. La fonction de répartition de la loi de Student à  $(N4 - p - 2)$  degré de liberté s'obtient sous Matlab avec le code suivant :

```
>> tcdf(T, N4 - p -2)
```

### Manipulation 9

- ➔ Calculer le vecteur  $\hat{\sigma}^2$ .
  - ➔ Calculer le vecteur  $T$ .
  - ➔ Calculer le vecteur de p-valeur.
  - ➔ Visualiser les p-valeurs en 3D en utilisant le masque et le code de la manipulation 7.
  - ➔ On peut seuiller les p-valeurs pour un seuil de signification  $\alpha = 0.01$  afin de visualiser les voxels pour lequel le contraste est le plus significatif.
  - ➔ Superposer la carte seuillée à une des acquisitions  $I(:, :, :, i)$  afin de visualiser les zones du cerveau activées par l'audition.
- 

## 3 Pour aller plus loin

- ⚡ Quelles sont les zones identifiées à l'aide du contraste précédent ?
- ⚡ L'analyse par T-test et l'analyse par corrélation donnent-elles le même résultat ? S'il vous reste du temps, vous pouvez implémenter l'une ou l'autre des méthodes afin de comparer les résultats.