

1 Modèle locaux

1.1 Prise en main de Mrtrix

Mrtrix est un logiciel de traitement des données de diffusion écrit en C++. Un guide d'installation et d'utilisation est disponible sur le site de Mrtrix dans les onglets correspondant.

Mrtrix se présente sous la forme de lignes de commandes bash. Ceci permet de réduire la consommation mémoire (et de ne pas avoir à développer une interface graphique). En outre vous bénéficiez des avantages du bash à savoir :

- *autocomplétion* : Taper le début d'une commande et appuyer sur la touche **Tab** pour obtenir les différentes possibilités
- *piping* : vous pouvez enchaîner deux commandes en liant les sorties de la première aux entrées de la seconde en utilisant le symbole `|` par exemple pour trouver l'ensemble des fichier contenant bubu dans un répertoire taper :

```
1 ls -l | grep *bubu*
```

L'inconvénient principal est de connaître un minimum de bash pour pouvoir survivre (se déplacer dans les dossiers etc.).

Comme pour FSL, la façon la plus efficace de se renseigner sur une commande est de taper la fonction dans le terminal avec l'option `-help` pour en obtenir une description détaillée. A noter que cette "documentation" est assez inégale, certaines commandes étant totalement documentées (options par défaut, paramètres possibles etc.) tandis que d'autres contiennent juste une description des paramètres. Pour comprendre plus précisément ce que font les algorithmes sous-jacent il est toutefois nécessaire de consulter les publications citées en références ou de se plonger dans le code C++ ce qui est instructif mais fastidieux.

Important : Mrtrix est paramétré par un (des) fichier(s) de configuration. Ce fichier concerne essentiellement des paramètres d'affichage a priori sans importance pour ce qui nous concerne. Cependant, certains paramètres (`BValueScaling`, `BZeroThreshold`) concernant par exemple le schéma d'acquisition peuvent influencer le reste des traitements. Pour plus de renseignement sur le fichier de configuration voir la documentation Mrtrix

Ouvrir un terminal et taper :

```
1 mrconvert -help
```

Ceci vous affichera la documentation de la commande `mrconvert`. Nous allons l'utiliser pour extraire et convertir les données obtenues lors des prétraitements afin de les utiliser dans Mrtrix. La commande `mrinfo` permet d'afficher de façon synthétique les informations contenues dans l'en-tête d'un fichier.

1.2 Préparation des données

Les données obtenues à l'issue des prétraitements sont situées dans le dossier `input`. Mrtrix dispose de formats natif non compressés le `.mif` `.mih`. Il supporte également le format `.nii` et `.nii.gz`. Pour la suite de ce TP, je vous propose d'utiliser le format `.mif` pour les données susceptibles d'être traitées (données de diffusion, masques, tenseurs, coefficients d'harmoniques sphériques) et le format `.niiou` `nii.gz` pour les grandeurs d'affichages (quantités dérivées essentiellement). Commençons la préparation des données.

1. Ouvrir le fichier `LR.bval` contenant les bvalues d'acquisition. Que remarque-t-on ? Pour palier à ce problème (ce n'en est pas un en soi mais le comportement des logiciels étant imprévisible mieux vaut anticiper) une version arrondie du fichier `bval` nommée `LR_rounded.bval` vous est fournie.
2. En utilisant la commande `mrconvert`, le fichier `LR_mrtrix.bvec` et le fichier `LR_rounded.bval` convertir le fichier `preprocessed_dwi_data.nii.gz` en un fichier `.mif` nommé par exemple, `preprocessed_dwi_data_rounded_scheme.mif`.
Prenez bien le fichier `LR_mrtrix.bvec` sinon vous aurez au moment de l'affichage une mauvaise surprise... les glyphes de tenseur ne seront pas dans la bonne orientation. Ceci est dû à `Mrtrix` qui considère que `FSL` fonctionne par défaut en `LAS`. Il réoriente donc le fichier `bvec` en inversant la première coordonnée et ce même si le fichier est déjà en `RAS`.
3. Faire de même avec le fichier `LR.bval` et générer un autre volume nommé par exemple `preprocessed_dwi_data_raw_scheme.mif`.
4. Dans le terminal taper :

```
1 mrview preprocessed_dwi_data_rounded_scheme.mif
```

Vous pouvez également taper uniquement la commande `mrview` et charger le fichier depuis l'interface graphique.

5. Explorer les fonctionnalités d'affichage (touches A,S,C) pour obtenir les vues axiale, sagittale et coronale), flèches haut / bas pour parcourir les coupes spatiales, flèches gauche/droite pour les dimension de diffusion. Dans l'onglet *Image/Propriétés*, vous retrouvez les informations contenues dans l'en tête du volume. Intéressez vous en particulier au schéma d'acquisition. Dans quel espace de coordonnées est-il exprimé ?
6. Convertir de même les autres fichiers dans le dossier `derived_input`. Le fichier `T1_brain_mask_diff_space.mif` que vous générez va en particulier nous servir pour toutes les opérations d'estimation des modèles.
7. Si vous souhaitez quand même lancer les calculs, allez dans l'onglet *Tools/ROI editor* et créez votre propre masque. Vous pouvez par exemple prendre un "cube" qui inclut le corps calleux et une partie du faisceau cortico spinal. Sauvegarder ce ROI. Vous pouvez à présent l'utiliser comme masque dans les calculs.

1.3 Un premier modèle, le tenseur de diffusion

1.3.1 Avec Mrtrix

Le script utilisé pour estimer le tenseur de diffusion est `dwi2tensor`.

1. Afficher

```
1 dwi2tensor -help
```

La méthode d'estimation fixée par le paramètre `-iter` influence-t-elle le tenseur obtenu et les indices dérivés ? Pour cela on a préalablement estimé le tenseur avec chacun des paramètres (compter 15 min pour estimer le tenseur avec le masque). Vous allez à présent créer les indices dérivés usuels en utilisant la commande `tensor2metric`.

2. Calculer la Fraction d'Anisotropie (FA), la Diffusivité Moyenne (MD), et le premier vecteur propre du tenseur au format nii en utilisant cette commande.

```
1 tensor2metric -adc MD.nii -fa FA.nii -vector ev.nii -num 1
```

1.3.2 Avec FSL

FSL permet également d'estimer les paramètres du tenseur de diffusion via la commande `dtifit`.

1. Afficher les options de cette commande

```
1 fsl5.0-dtifit -help
```

2. Lancer l'estimation du tenseur restreint au masque du cerveau en utilisant la méthode des moindres carrés classique.
3. Ouvrir les indices dérivés obtenus dans votre logiciel de visualisation préféré (Mrview ou Fslview) et comparer. Retrouve-t-on exactement les mêmes valeurs ? Regarder en particulier la FA et la MD.

1.4 Un autre modèle, la Déconvolution Sphérique Contrainte (CSD) et variantes

Comme vu en cours, la CSD est une méthode faisant partie de la famille des déconvolutions sphérique. Le *signal* mesuré est supposé provenir d'un mélange continu d'une fonction signal de base (signal type d'une population de fibre) et d'une fonction d'orientation. A noter, et nous le verrons dans la suite de ce TP, qu'il existe de nombreuses variantes et implémentations (et variantes d'implémentation) de cette technique. A noter également que si cette méthode permet d'augmenter la résolution angulaire ceci se fait au prix de l'ajout d'apriori.

1.4.1 Estimation de la réponse impulsionnelle

Afin de lancer l'algorithme de déconvolution sphérique il nous faut premièrement estimer / fixer la réponse impulsionnelle qui sera utilisée par la suite.

Pour cela on utilisera la commande `dwi2response`.

1. Explorer les options du script `dwi2response`

Comme indiqué dans la documentation du logiciel, la plupart des méthodes d'estimation de la réponse impulsionnelle fonctionne de manière similaire :

1. sélectionner un ensemble de voxels ne contenant qu'une seule population de fibre
2. estimer le signal moyen de cette population en utilisant un modèle simple (ex tenseur ou `csd`) . Est-ce pertinent ?

Vous trouverez sur cette page la documentation détaillée générale de chacune des méthodes disponible.

L'estimation des reponses impulsionnelles est assez longue, c'est pourquoi les réponses ont été précalculées. Néanmoins vous pouvez créer votre propre réponse impulsionnelle via l'option `manual`.

1. Ouvrez dans mrview le contraste qui vous semble le plus pertinent, par exemple une carte de FA, la T1 dans l'espace de diffusion.
2. A l'aide de l'outil ROI de mrview (Tools/Roi Editor) ou Ctrl + F3 dessiner une ROI correspondant à une zone ou vous pensez que les fibres ne possèdent qu'une direction (par exemple dans le corps calleux)
3. Sauvegarder votre ROI et passez la comme argument au script dwi2response.

Les réponses impulsionnelles sont sauvegardées sous forme de coefficients d'harmoniques sphérique, une base fonctionnelle très utilisée en diffusion. Vous pouvez visualiser les réponses impulsionnelles en utilisant la commande shview.

1. A l'endroit où vous avez enregistré le fichier taper

```
1 shview monfichier.txt
# option plus generale shview chemin_relatif_mon_fichier
```

Nous avons à présent tout ce qu'il faut pour faire de la déconvolution. Ceci se fait via la commande `dwi2fod`.

1. Comme précédemment taper la commande dans le terminal et explorer les options.

Remarque : Après consultation des forums Mrtrix et de la FAQ dédiée (notez au passage que les données du connectome font l'objet d'une section spéciale) il s'avère que le comportement des deux options fournies (`csd`, `msmt_csd`) ont des attentes un peu différentes. La première ne prend en compte qu'une seule bvalue non nulle (nous avons bien fait de les arrondir dans le doute) et ne s'occupe pas du $b=0$. Tandis que la méthode `msmt_csd` considère que $b=0$ est une couche comme une autre et l'utilise pour l'estimation. A noter que le nombre de bvalues différentes passées détermine le nombre de paramètres que vous pouvez estimer. Ainsi en passant une simple bvalue non nulle et des b_0 vous pouvez obtenir deux . Au delà de ces considérations pratiques, les deux commandes (et les algorithmes sous-jacents sont différents).

1. Explorer les (nombreuses) options du choix `csd` simple

Vous pouvez constater que l'algorithme `csd` possède de nombreuses option de régularisation pour l'estimation de la FOD (ou plus justement des coefficients de sa décomposition en harmonique sphérique). Le problème d'estimation de ces coefficients est un problème d'optimisation sous contrainte. Dans cette première version de l'algorithme les contraintes sont "faibles". Dessin + explication de la contrainte de positivité des valeurs et d'intégrale unitaire.

Le modèle `msmt_csd` implémente apparemment (je n'ai pas eu le temps de vérifier exactement dans le code C++) une contrainte "dure", problème de minimisation sous contrainte. Il semblerait qu'une bonne pratique soit de toujours utiliser `msmt_csd` même sans faire de classification de tissu et sans multishell ne serait ce que pour bénéficier de la contrainte en dur.

D'après les discussions sur les forums Mrtrix, il est semble-t-il recommandé d'utiliser la méthode hollander en combinaison avec le `msmt_csd`.

2. Visualiser les FOD obtenues. Pour cela procéder en deux temps.
3. Charger les coefficients des FOD comme un volume et regarder le premier élément du volume 4D . Comment pouvez vous l'interpréter ? Rappel : l'objet que vous avez décomposé est une densité de probabilité d'orientation de fibres.
4. Fermer ce volume et charger par exemple un volume de FA ou le volume T1 dans l'espace de diffusion.

5. Charger les FOD en utilisant l'onglet ODF. Jouer avec les options. En particulier, pour la csd simple activer et désactiver l'affichage des valeurs négatives. Que remarque-t-on ?

1.5 Et avec FSL ?

Hormis la fonction DtiFit, FSL ne dispose pas de fonction d'estimation directe. (en réalité si mais de façon détournée). Selon la version de FSL dont vous disposez vous aurez accès à différents modèles. L'estimation des modèles dans FSL se fait par estimation bayésienne et méthode de MonteCarlo (tirage selon la distribution a priori des paramètres pour estimer la distribution a posteriori des paramètres du modèle). Cette façon de procéder est "rigoureuse" car on marginalise la distribution de probabilité (pas d'estimation ponctuelle). Cependant elle est extrêmement gourmande en ressources (temps de calcul, RAM). Les temps de traitements explosent sur un sujet du HCP. Le sujet lancé pour la formation a pris 2 jours (pour 36 coupes sur 110) sur un cluster de calcul sans GPU. Les modèles disponibles via cette méthode sont :

- Tenseur
- Ball and Stick
- Extensions du Ball and Stick

Pour plus d'information consulter la documentation FSL de Bedpostx (Attention le plus sûr est d'afficher la documentation en ligne de commande le wiki FSL n'étant pas forcément à jour) Vous pouvez également utiliser un modèle de type signal, le qball (ces différentes versions) Attention la préparation des fichiers est un peu (très) fastidieuse et doivent respecter des conditions bien précises. (Il faut réorganiser entièrement le volume des données par shell de bvalues non nulle ascendante et incorporer au moins 1 b0 dans chaque shell) Le temps de calcul est encore plus long que pour bedpostx. A noter qu'il est possible de faire de l'estimation par fitting simple en ne demandant qu'une itération. Une version parallélisée de `q_boot` `q_boot_parallel` est proposée par FSL mais la syntaxe est légèrement différente et se devine plus qu'elle n'est expliquée.