

DMRI データ処理手順説明書

ATR Neural Information Analysis Laboratories(Dec. 1, 2017)

内容

1.はじめに	1
2.動作環境、OS.....	1
3.必要なソフトウェア	1
4.PC とソフトウェア	2
5.チュートリアルデータ	4
6.実行準備	4
ソフトウェア設定	4
MRTrix の設定ファイルを自分の Linux Home ディレクトリ下にコピーする	4
解析の開始	4
7.MRI data preprocessing.....	5
1.DICOM 形式から NIfTI(.nii)形式に変換する.....	5
2.NIfTI 形式構造画像を SPM8 によりバイアス補正をかける	5
3.NIfTI 形式の構造画像を FSL 処理形式(.nii.gz)に変換、脳画像・脳マスク画像を抽出する	5
8.dMRI data preprocessing	7
1.DWI-DICOM 画像から DWI 画像を取り出す	7
2.DWI 画像にモーション補正をかける	7
3.DWI 画像から脳画像・脳マスク画像を抽出する	8
4.DWI 画像から位置合わせ用 FA 画像(Fractional anisotropy)を作る	9
9.Coregistration.....	9
1.皮質形状・領野情報を、FreeSurfer を使って抽出する dmri_process_T1_freesurfer(p);	9
2.画像空間位置合わせ処理	9
3.FA 画像周囲のノイズを除去する	10
10.Parcellation & ROI generation	12
1.皮質モデル取り込み処理	12
2.皮質区分化(Parcel)処理.....	12
3.区分化情報に基づいて FA 空間の Volume ファイルを作る	12
4.Fiber Tracking を行うエリアマスク Volume ファイルを作る.....	13
11.Estimate fiber orientation distribution	13
12.Tracking fibers	15
1.ファイバートラッキングを実行する	15
2.解剖学的結合パラメータを計算し、Diffusion MRI file を作る.....	15
更新履歴	18

1.はじめに

本ドキュメントは、DMRI(Diffusion MRI)データから **Connectivity Matrix** を求めるためのプログラムの環境設定・チュートリアルの使用手順に関する説明を行うものである。事前にインストールが必要なソフトウェアについては、「3.必要なソフトウェア」を参照のこと。

2.動作環境、OS

OS:Linux 64bit

3.必要なソフトウェア

1. MRTrix 0.2.10 or later

FiberTracking を行う。

<http://jdtournier.github.io/mrtrix-0.2/index.html>

<https://www.nitrc.org/projects/mrtrix/>

2. FSL 4.1 or later

画像処理、変換等に使用。

<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>

3. SPM8

構造画像のバイアスコレクションに使用。

<http://www.fil.ion.ucl.ac.uk/spm/software/spm8/>

4. FreeSurfer 4.5 or later

脳皮質形状抽出に使用。

<https://surfer.nmr.mgh.harvard.edu/>

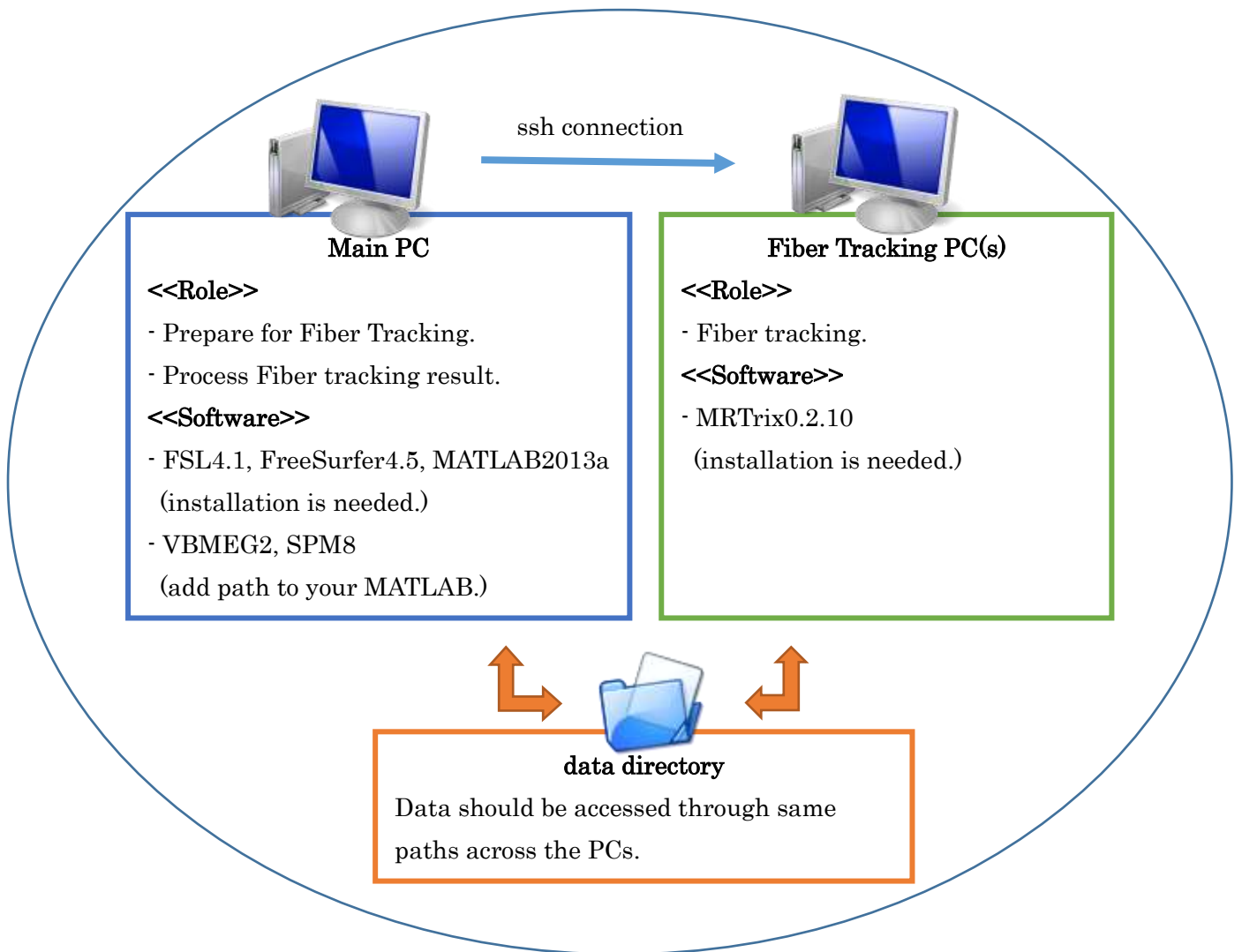
5. MATLAB

動作確認は 2013a で行っている。

6. VBMEG 2.0

以下、プログラムディレクトリを\$VBMEG と表記する。

4.PC とソフトウェア



- Main PC は 1 台用意する。
- FiberTracking PC は並列計算のために複数台用意する。
- Main PC と Fiber Tracking PC は、同じファイルパスでデータ共有出来るようにする。
- MainPC と FiberTrackingPC の SSH 接続はパスワード無し行えるようにする(次ページに方法を記載している)。

4.1 PC 間をパスワード無しで SSH 接続するための設定

Linux ターミナルで下記の太字部分を実行する(ユーザー:rhayashi での例)

```
$ ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/cbi/rhayashi/.ssh/id_rsa):
```

```
[ENTER]
```

```
Created directory '/home/cbi/rhayashi/.ssh'.
```

```
Enter passphrase (empty for no passphrase):
```

```
[ENTER]
```

```
Enter same passphrase again:
```

```
[ENTER]
```

```
Your identification has been saved in /home/cbi/rhayashi/.ssh/id_rsa.
```

```
Your public key has been saved in /home/cbi/rhayashi/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
b3:5f:2a:7f:cb:42:dd:ae:bb:cf:ab:cd:ef:gh:ij:96
```

```
$ cd /home/cbi/rhayashi/.ssh
```

```
$ cat id_rsa.pub >> authorized_keys
```

その後、FiberTracking を行う全てのホストに対して、ssh 接続でログインし、known hosts に追加する。例えば、ホストが cbi-node20, 21, 22 なら、ssh コマンドは合計 3 回実行する。

```
_rhayashi@myhost<1> ssh cbi-node20
```

```
The authenticity of host 'cbi-node20 (xxx.xxx.xx.xxx)' can't be established.
```

```
RSA key fingerprint is e8:f6:5e:94:b5:af:b8:70:85:7f:ab:cd:ef:gh:ij:c0.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'cbi-node20,xxx.xxx.xx.xxx' (RSA) to the list of known hosts.
```

5. チュートリアルデータ

以下のデータがサンプルデータである。

/home/cbi-data5/DTI/sample/3D	T1 構造画像(DICOM 形式)
/home/cbi-data5/DTI/sample/DTI	DMRI データ(DICOM 形式)

チュートリアルの処理結果格納ディレクトリ

/home/cbi-data5/DTI/processed

6. 実行準備

ソフトウェア設定

\$VBMEG/vbmegrc を編集する

```
# for FreeSurfer
FREESURFER_HOME=/home/cbi-data20/common/software/external/freesurfer/freesurfer4.5.0
# for FSL
FSLDIR=/home/cbi-data20/common/software/external/fsl/fsl4.1
# for MRtrix
MRTRIXDIR=/home/cbi-data20/common/software/external/mrtrix/mrtrix0.2
```

MRtrix の設定ファイルを自分の Linux Home ディレクトリ下にコピーする

dmri_setenv.m を実行する。

\$>dmri_setenv

例 : /home/cbi/rhayashi/.mrtrix.conf

Analyse.LeftToRight: false
NumberOfThreads:8

NumberOfThreads は、FiberTracking を行う時のスレッド数。ホストのコア数を超えないように設定する。

解析の開始

1. Matlab を起動する

\$>/home/cns/matlab/r2013a/bin/matlab

2. VBMEG ディレクトリに移動し、vbmeg.m を実行する

(以下は MATLAB 上での操作)

cd \$VBMEG

vbmeg;

3. SPM8 のパスを MATLAB に追加する

4. \$VBMEG/functions/tool_box/dmri_processor/dmri_set_parm_tutorial.m を編集し、T1(DICOM 形式)データディレクトリ、DMRI (DICOM 形式) ディレクトリ、出力先ディレクトリを指定する

```
=====
p.t1_dicom_dir      = '/home/cbi-data5/DTI/sample/3D'  % T1-DICOM
p.dmri_dicom_dir    = '/home/cbi-data5/DTI/sample/DTI'; % DMRI-DICOM
p.output_dir        = '/home/cbi-data5/DTI/processed';  % 出力先
=====
```

5. パラメータをロードする

```
p = dmri_set_parm_tutorial;
```

6. 以下、次章以降の処理手順に従って処理を行う

\$VBMEG/functions/tool_box/dmri_processor/dmri_tutorial.m に全処理ステップを記載しているので参考のこと

7.MRI data preprocessing

1.DICOM 形式から NIfTI(.nii)形式に変換する

```
dmri_process_T1_dicom_convert(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/struct/Subject.nii
```

2.NifTI 形式構造画像を SPM8 によりバイアス補正をかける

```
dmri_process_T1_bias_correct(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/struct/mSubject.nii
```

3.NifTI 形式の構造画像を FSL 処理形式(.nii.gz)に変換、脳画像・脳マスク画像を抽出する

```
dmri_process_T1_brain_extract(p);
```

閾値(p.t1_brain_threshold)を 0~1.0 の間で設定する。数字が小さいほど抽出エリアが増える。実行が終わると fslview が起動する。確認しながら、良い値を探す。

<<出力ファイル>>

```
$OUTPUT_DIR/struct/struct.nii.gz
```

```
$OUTPUT_DIR/struct/struct_brain.nii.gz (脳画像)
```

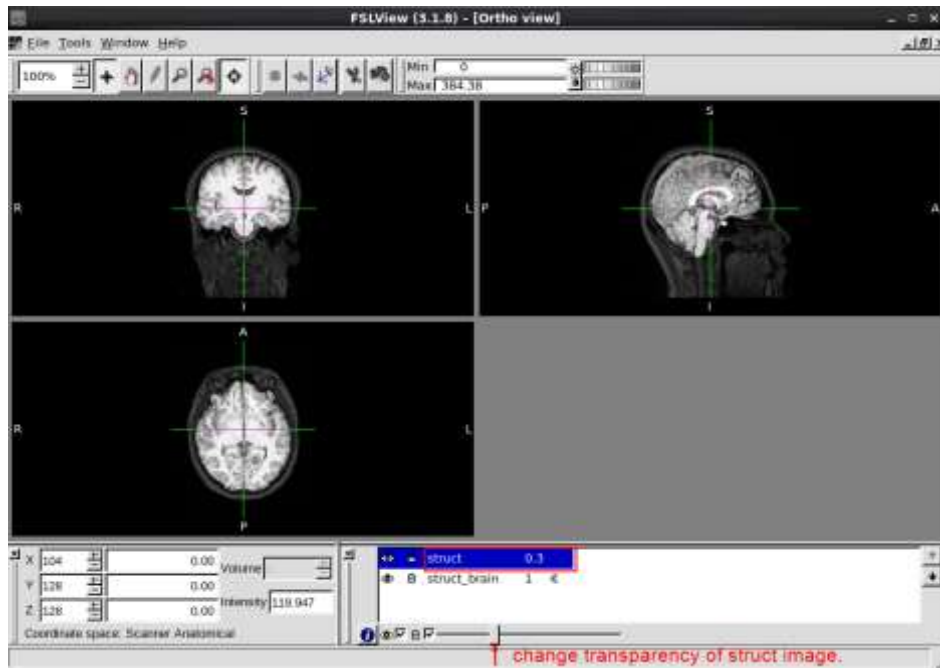
```
$OUTPUT_DIR/struct/struct_brain_mask.nii.gz(脳マスク画像)
```



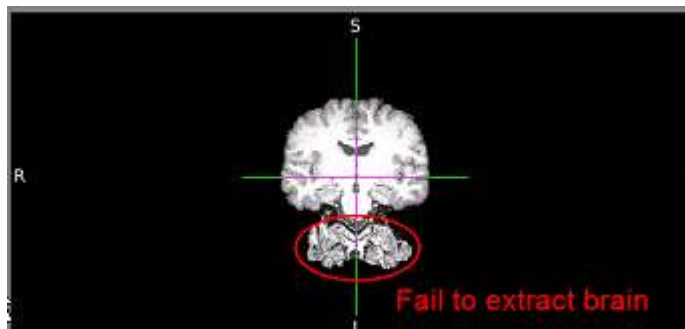
<<確認ポイント>>

```
dmri_system('fslview $OUTPUT_DIR/struct/struct_brain.nii.gz  
$OUTPUT_DIR/struct/struct.nii.gz')
```

脳部位抽出の成否を確認。構造画像にフォーカスを当て、画面下部の透過度をスライダーで変化させながら確認する。



失敗例



8.dMRI data preprocessing

1.DWI-DICOM 画像から DWI 画像を取り出す

```
dmri_process_DWI_import(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/DWI/data_dwi.nii.gz
```

```
$OUTPUT_DIR/DWI/data_dwi.bvec
```

```
$ OUTPUT_DIR/DWI/data_dwi.bval
```

2.DWI 画像にモーション補正をかける

```
dmri_process_DWI_motion_correct(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/DWI/data_dwi_m.nii.gz
```

```
$OUTPUT_DIR/DWI/data_dwi_m.bvec
```

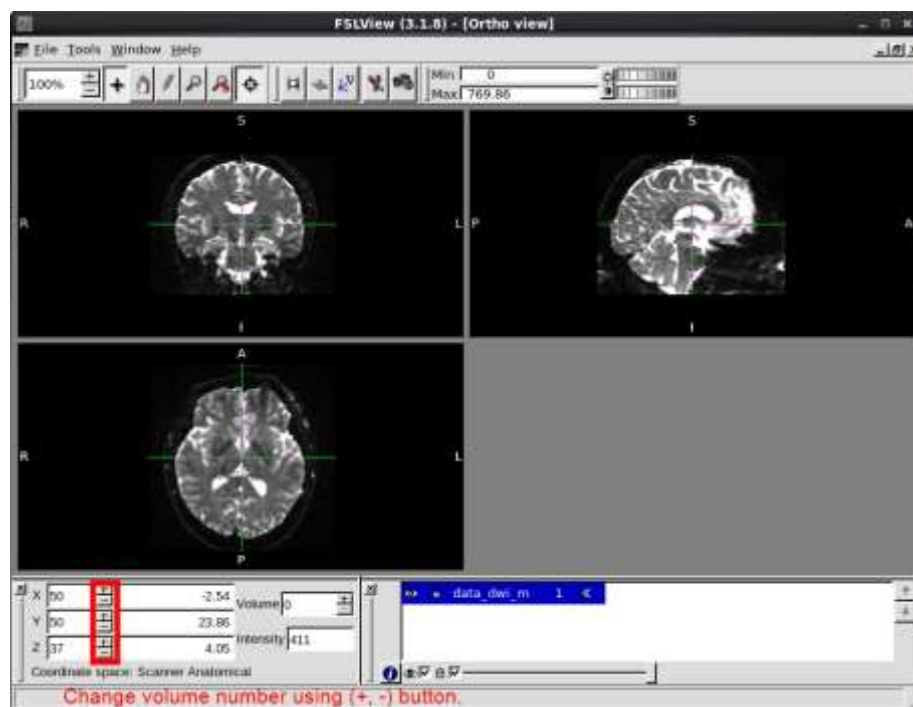
```
$OUTPUT_DIR/DWI/data_dwi_m.bval
```

🔍 <<確認ポイント>>

コマンド : MATLAB 上で外部コマンド実行

```
dmri_system('fslview $OUTPUT_DIR/DWI/data_dwi_m.nii.gz
```

各ボリューム同士の上に大きな位置ずれがないか確認する。



3.DWI 画像から脳画像・脳マスク画像を抽出する

```
dmri_process_DWI_brain_mask_extract(p);
```

閾値(p.dmri_brain_threshold)を画像に応じて 0~1.0 の間で設定する。数字が小さいほど抽出エリアが増える。実行が終わると `fslview` が起動するので、確認しながら良い値を探す。

<<出力ファイル>>

```
$OUTPUT_DIR/DWI/data_dwi_m_brain.nii.gz
```

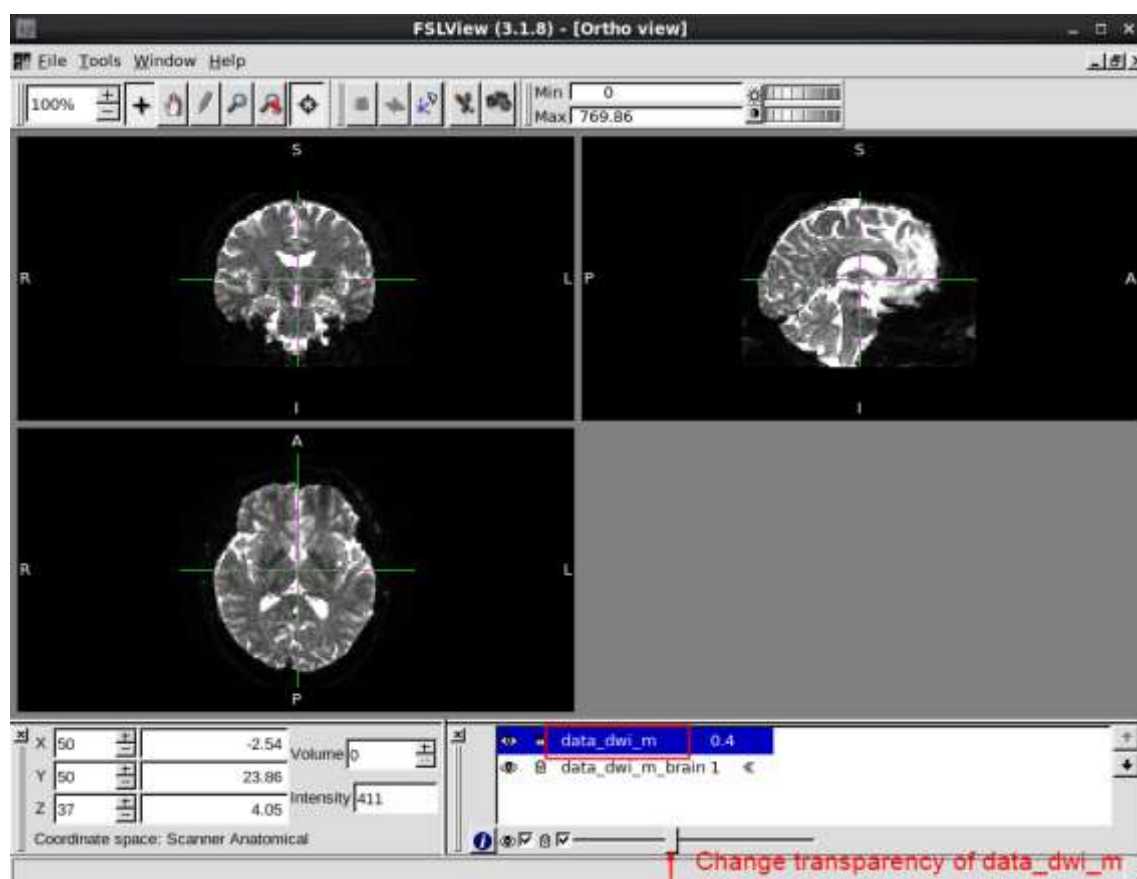
```
$OUTPUT_DIR/DWI/data_dwi_m_brain_mask.nii.gz
```

🔴 <<確認ポイント>>

MATLAB 上でコマンド実行。

```
dmri_system('fslview $OUTPUT_DIR/DWI/data_dwi_m_brain.nii.gz  
$OUTPUT_DIR/DWI/data_dwi_m.nii.gz');
```

脳部位抽出の成否を確認。モーションコレクション済み DWI 画像にフォーカスを当て、画面下部の透過度をスライダーで変化させながら確認する。



4.DWI 画像から位置合わせ用 FA 画像(Fractional anisotropy)を作る

```
dmri_process_FA_image_coreg_create(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/DWI/data_FA_coreg.nii.gz
```

9.Coregistration

1.皮質形状・領野情報を、FreeSurfer を使って抽出する

```
dmri_process_T1_freesurfer(p);
```

※ 1 2 ～ 2 4 時間程度かかる

* すでに作成済の FreeSurfer データが手元にある場合は、
FreeSurfer 被験者ディレクトリ: p.ref_fs_dir を指定する。

<<出力ディレクトリ>>

```
$OUTPUT_DIR/freesurfer/Subject
```



<<確認ポイント>>

MATLAB 上でコマンド実行。

```
dmri_freesurfer_surf_view('$OUTPUT_DIR/freesurfer/Subject');
```

抽出した surface の確認を行う。皮質と構造画像があっているか確認する。

2.画像空間位置合わせ処理

```
dmri_process_images_coregister(p);
```

FreeSurfer 空間、構造画像空間、FA 画像空間の位置合わせを行う。

<<出力ディレクトリ>>

空間変換情報を含む以下のディレクトリが作成される。

```
$OUTPUT_DIR/transwarp_info
```

3.FA 画像周囲のノイズを除去する

```
dmri_process_FA_image_clean(p);
```

T1 構造画像マスクを用いて FA 画像周囲のノイズを除去する。

<<出力ファイル>>

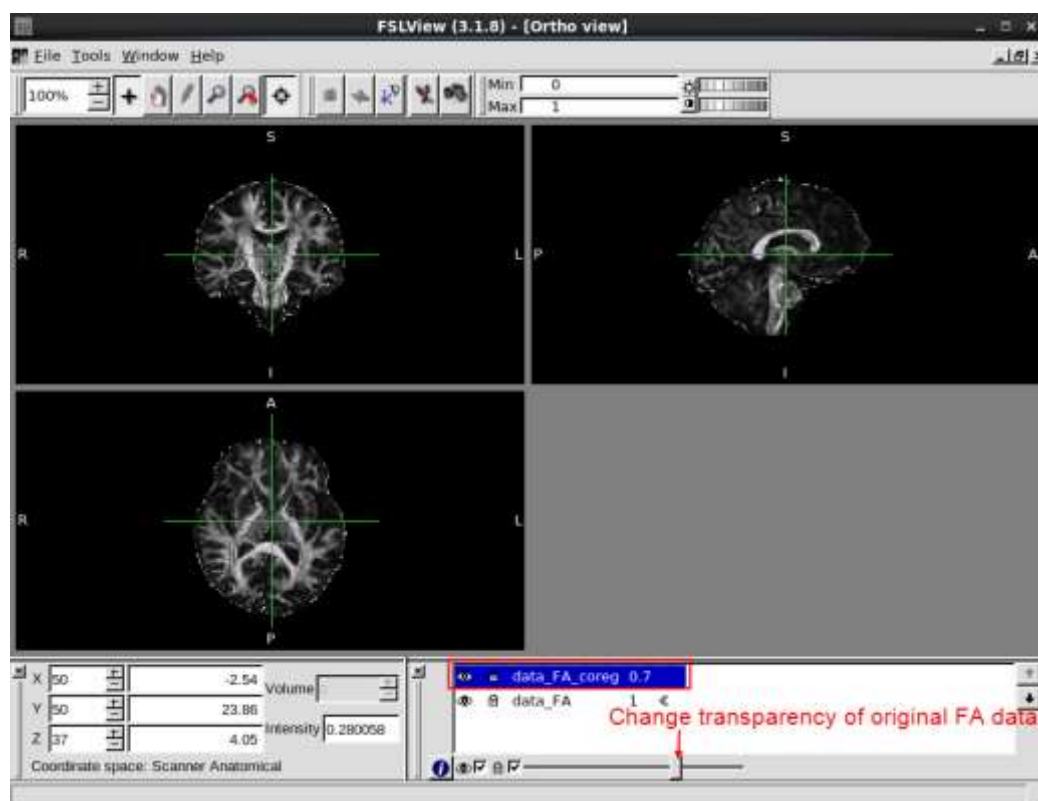
```
$OUTPUT_DIR/DWI/data_FA.nii.gz
```

🔍 <<確認ポイント 1 >>

MATLAB 上で外部コマンド実行(tutorial に記載あり)

```
dmri_system('fslview $OUTPUT_DIR/DWI/data_FA.nii.gz  
$OUTPUT_DIR/DWI/data_FA_coreg.nii.gz');
```

FA 画像周囲の白い点状ノイズが取り除かれているかどうか確認する。元画像とノイズ除去後の画像を表示させ、元画像(data_FA_coreg.nii.gz)を選択し、スライダーを左右に移動して、透過度を変化させてノイズの除去状況を確認する。

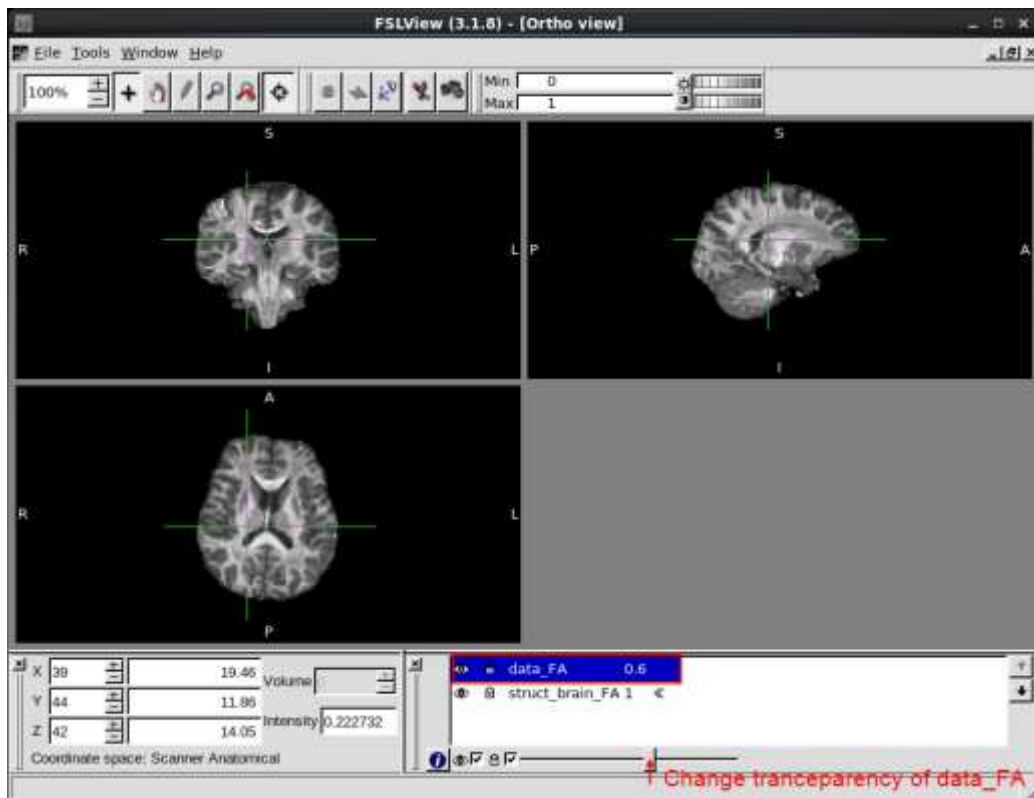


🔍 <<確認ポイント 2>>

MATLAB 上で外部コマンド実行(tutorial に記載あり)

```
dmri_system('fslview $OUTPUT_DIR/DWI/struct_brain_FA.nii.gz  
            $OUTPUT_DIR/DWI/data_FA.nii.gz');
```

構造画像と FA 画像を表示し、位置合わせが正しく行われていることを確認する。FA 画像を選択し、画面下部のスライダーを左右に動かして透過度を変化させ、2つの画像の重なり具合を見る。



<<トラブルシューティング>>

T1 の脳抽出がうまくいかない場合、うまく FA 画像周囲のノイズを取り除けないことがある。その場合、以下の MRtrix の関数を使用したコマンドを利用して FA 画像周囲のノイズを取り除く (tutorial の中にコマンドを記載している)。

```
dmri_system('erode data_dwi_m_brain_mask.nii - | erode - - | erode - - | mrmult  
data_FA_coreg.nii - data_FA.nii');
```

 (事前に.nii.gz を gunzip して.nii にする必要あり。)

(上記コマンドの実行により data_FA.nii.gz を作成する場合には、DWI 画像処理、3.DWI 画像から脳画像・脳マスク画像を抽出する、で正しく DWI の脳領域が抽出されている必要がある。)

10. Parcellation & ROI generation

1. 皮質モデル取り込み処理

```
dmri_process_cortex_import(p);
```

FreeSurfer の皮質モデルを VBMEG で取り込む。

* すでに VBMEG でインポート済の脳モデルファイルがある場合は、

FreeSurfer 被験者ディレクトリ	: p.ref_fs_dir
VBMEG 皮質モデルファイル(.brain.mat)	: p.ref_brain_file
VBMEG 領野ファイル(_aal.area.mat)	: p.ref_aal_area_file

を指定してください。

<<出力ファイル>>

```
$OUTPUT_DIR/vbmeg/Subject.brain.mat(このファイルのみ使用)
```

```
$OUTPUT_DIR/vbmeg/Subject.area.mat
```

```
$OUTPUT_DIR/vbmeg/Subject.act.mat
```

```
$OUTPUT_DIR/vbmeg/Subject_aal.area.mat
```

```
$OUTPUT_DIR/vbmeg/Subject_atlas.area.mat
```

```
$OUTPUT_DIR/vbmeg/Subject_brodmann.area.mat
```

2. 皮質区分化(Parcel)処理

```
dmri_process_cortex_parcel(p);
```

2000 点の頂点を選び出し、そこから皮質を区分化する。parcels.mat に結果を保存する。

<<出力ディレクトリ>>

```
$OUTPUT_DIR/vbmeg/fs_info
```

<<出力ファイル>>

```
$OUTPUT_DIR/vbmeg/parcels.mat
```


3. 区分化情報に基づいて FA 空間の Volume ファイルを作る

```
dmri_process_FA_parcel_volume_files_create(p);
```

<<出力ファイル>>

```
$OUTPUT_DIR/parcels/
```

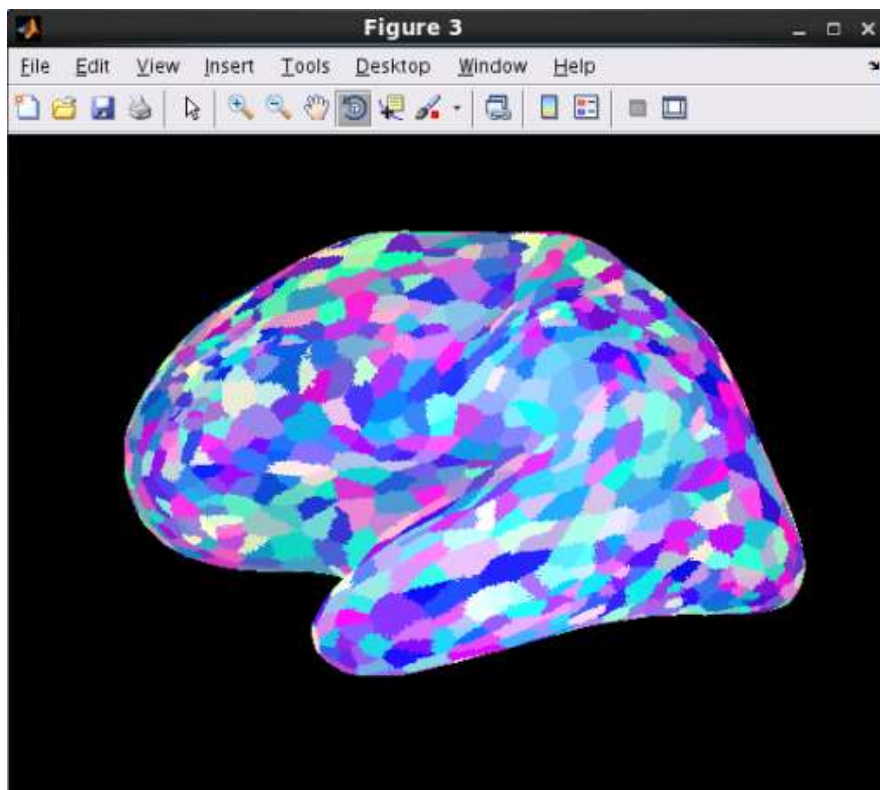
- parcel_all.nii.gz(全ての Parcel を含む Volume file)
- parcel1-N.nii.gz(Parcel 毎の Volume file)
- parcel_ex1-N.nii.gz(全ての Parcel から Parceln だけを除いた Volume file).
- parcel_mm_coord.mat(parcel を FA 空間の mm 座標系で保存)

 <<確認ポイント>>

```
dmri_parcel_view($OUTPUT_DIR/freesurfer/Subject, $OUTPUT_DIR, 'l'); %left
```

```
dmri_parcel_view($OUTPUT_DIR/freesurfer/Subject, $OUTPUT_DIR, 'r'); %right
```

で下記のような Parcel を表示することができる。極端におかしい場所がないかどうか確認する (File->close で閉じる)。



4.Fiber Tracking を行うエリアマスク Volume ファイルを作る

(エリアマスク=白質+全ての Parcel を含むエリア)

```
dmri_process_fiber_tracking_mask_create(p);
```

<<出力ファイル>>

\$OUTPUT_DIR/fibertrack/wm.nii.gz(白質)

\$OUTPUT_DIR/fibertrack/parcel_wm.nii.gz (エリアマスク)

11.Estimate fiber orientation distribution

```
dmri_process_fodf_calc(p);
```

※この処理は MRTrix を使用する為、MRTrix が動作するホスト上で行う。

<<出力ファイル>>

\$OUTPUT_DIR/fibertrack/CSD6.mif

\$OUTPUT_DIR/fibertrack/response.txt

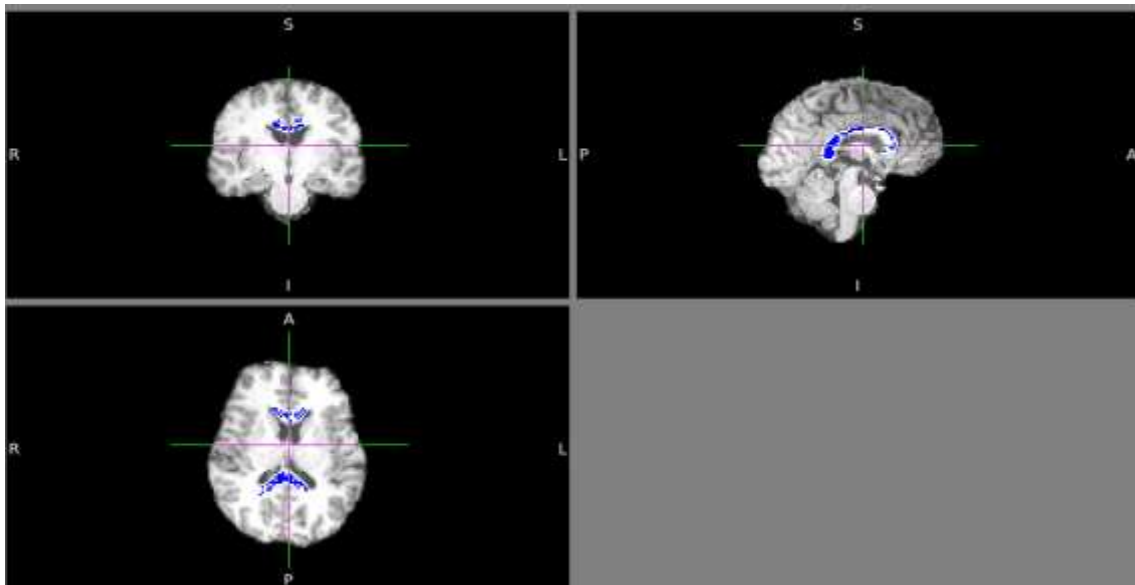
\$OUTPUT_DIR/fibertrack/max_cluster_val

\$OUTPUT_DIR/fibertrack/sf.nii.gz

🔥 <<確認ポイント 1>>

コマンド：MATLAB 上で外部コマンド実行

```
dmri_system('fslview $OUTPUT_DIR/DWI/struct_brain_FA.nii.gz  
$OUTPUT_DIR/fibertrack/sf.nii.gz -l Blue');
```



1は小文字のエル。脳梁付近が青く表示されていれば OK。T1 の脳抽出に失敗していると、脳以外の場所が青くなっている場合がある(この場合は、9.3.のトラブルシューティングを試す)。

🔥 <<確認ポイント 2>>

1. MATLAB 関数を実行する(tutorial に記載あり)

```
dmri_mrview($OUTPUT_DIR/DWI/data_FA.nii.gz);
```

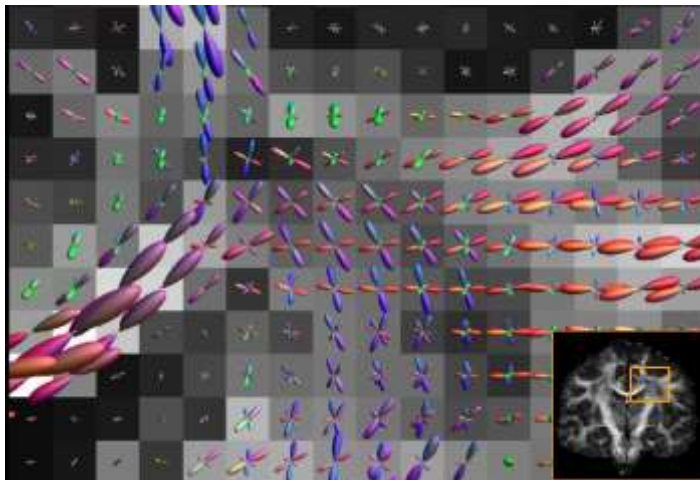
ツール：<http://jdtournier.github.io/mrtrix-0.2/general/mrview.html>

2. ツールが起動するので、F9 を押す。画面右側にダイアログが表示されるので、ダイアログ情報の **tactography** となっている選択ボタンを **orientation plot** に変更する。

3. **browse** ボタンから\$OUTPUT_DIR/fibertrack/CSD6.mifを選択し、**overlay** にチェックを入れる。各ボクセルに対して **Fiber** 方向密度関数がプロットされる。

※プロットが遅いので、Ctrl+マウスホイールを用いて全体を拡大するとよい。脳梁と皮質脊椎路の交点で複数の方向を持った密度関数が、大体以下の図 (Tournier, OHBM

2012 educational course slides) のように推定されているかどうかを確認。



12.Tracking fibers

1.ファイバートラッキングを実行する

```
dmri_process_fiber_tracking_execute(p);
```

この関数を実行すると p.host で指定されたホスト上で FiberTracking が行われる。

MRTrix が各ホスト上で独立に実行される。NumberOfThreads の設定が、ホストのコア数を超えないように設定すること。

また、FiberTracking を行った結果、出力される TCK-file を処理するために、p.Nworker で指定された数の MATLAB プロセスが、本関数を実行中のマシン上で起動する。こちらも本関数実行中のマシンのコア数を超えないように設定すること。

<<出力ファイル>>

\$OUTPUT_DIR/fibertrack/anat_matrix.mat

<<実行を中断したい場合>>

ctrl+c を押してプロセスを停止後、\$OUTPUT_DIR/fibertrack/の中に作成されている tpf3f1c622_27a1_43d7_b24d_51ef9c223644 のような、テンポラリディレクトリを削除する。処理が完全に終わっていない場合、削除できないことがあるが、時間を置いて何度かトライすれば削除することができる。

2.解剖学的結合パラメータを計算し、Diffusion MRI file を作る

```
dmri_process_connection_calc(p);
```

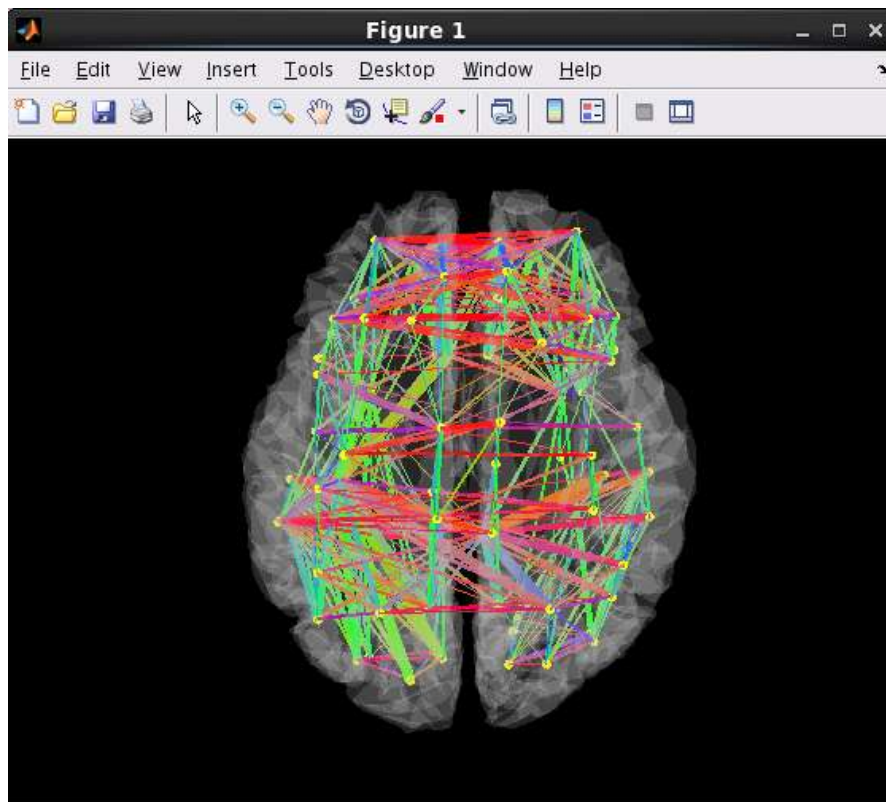
<<出力ファイル>>

\$OUTPUT_DIR/connectivity/connectivity.dmri.mat

🔍 <<確認ポイント>>

以下のようなコネクションが表示される(500 本程度)。脳全体に接続があるかなどを確認す

る。(事例研究が進んでいないため、どのようになれば正しいというのは、現時点(2015/8/28)では結論が出ていない。しかし、あまりに偏りが無いかなどを見ると良いと思われる)



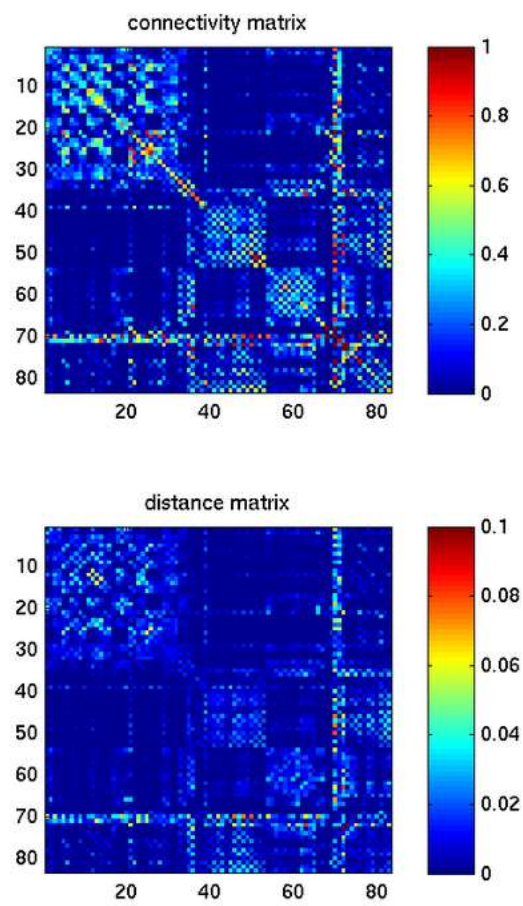


図 領野間の connectivity matrix、線維の距離行列

更新履歴

2015/6/3 章立てを変更。DTI を DMRI に文言変更。

2015/7/17 プログラム関数名称に伴う変更(dti_系列から dmri_系列への変更)

2015/8/28 tutorial_connection_calc.m の確認ステップを追加した。

2015/9/9 tutorial_connection_calc.m の結果に、領野間の connectivity matrix, 繊維の距離行列の図を追加した。

2016/7/29 VMEG_HOME, SPM8_HOME のパスを公式なものに変更した。

2016/8/31 erode コマンドの記述を修正した。

構造画像に NIfTI 形式を用いるように修正した。

SPM normalization file の作成ステップを追加した。

皮質モデルインポート(オプション)の説明を追加した。

2016/10/3 外部実行コマンド(!)を dmri_system 関数に置き換えた。

起動方法の変更(端末の環境変数設定が不要になった)

2016/10/31 関数体系を dmri_process_*.m に変更

2016/11/16 4.PC とソフトウェア

4.1 PC 間をパスワード無しで SSH 接続するための設定を追加。

2017/1/11 SSH 接続を FiberTracking hosts に行うことを追記。

2017/1/31 SPM normalization file の作成ステップを除去した。

2017/3/2 NumberOfThreads, p.Nworker の説明を追加した。

2017/6/30 ref_fs_dir, ref_brain_file, ref_aal_area_file について追記した。

2017/7/21 PC 間をパスワード無しで SSH 接続するところをわかりやすく変更した。