Neurodebian も利用した Ubuntu12.04 ファミリーへの Connectome mapper のインストール *

根本清貴 † 神谷昂平 ‡

2014 年 3 月 11 日

目次

1 Ubuntu のインストール 2

2 Neurodebian リポジトリの準備 2

3 apt-get を用いてインストールできるソフトウェア及びライブラリのインストール 3
- 3.1 Connectome Viewer . . . 3
- 3.2 FSL . . . 3
- 3.3 MRtrix . . . 4
- 3.4 Java . . . 4
- 3.5 Lapack . . . 4
- 3.6 Connectome mapper に必要なライブラリのインストール . . . 4
- 3.7 git のインストール . . . 5

4 apt-get ではインストールできないソフトウェアのインストール 5
- 4.1 Diffusion Toolkit . . . 5
- 4.2 Freesurfer . . . 6
- 4.3 Camino . . . 7
- 4.4 Camino-Trackvis converter . . . 8
- 4.5 MITK . . . 8
- 4.6 MITK Diffusion . . . 9
- 4.7 パスの設定 . . . 10
- 4.8 .bashrc の確認 . . . 10
- 4.9 新しいパスの読み込み . . . 11

* このドキュメントは、包括型脳科学研究推進支援ネットワークの活動の一環として作成しました。ドキュメントの作成にあたり、貴重なご意見をいただいた国立精神・神経医療研究センター病院放射線科下地啓五先生に深謝いたします。

† 筑波大学医学医療系精神医学

‡ 順天堂大学放射線科

4.10 Nipype のインストール 12

5 Connectome mapper **のインストール** 12

5.1 Connectome mapper の起動 12

6 **動作検証** 12

6.1 サンプルデータの入手 13

6.2 データの選択とチェック 13

以下の方法は、http://www.cmtk.org/mapper/installation.html を元に作成したものです。Ubuntu に特化していることと、Neurodebian リポジトリを利用するため、apt-get で入手できるものを先に記載するなど、改変しています。なお、作業は主にターミナルでうつ作業と、ホームディレクトリの.bashrc ファイルの編集になります。区別がつきやすいように、ターミナルでうつ作業は、

```
$ このように見えるのはすべてターミナルからの入力です
```

となり、.bashrc ファイルの編集は、

```
#ここに書いてある内容を.bashrc に記載します
#ソフトウェアのパス
SOFT=/usr/local/hogehoge
```

のようになっています。
なお、筆者が配布している Lin4Neuro は、20140311 版以降は、Freesurfer と Diffusion Toolkit のみをインストールすれば、Connectome mapper が立ち上がるように設定されています。したがって、Lin4Neuro の方は、Diffusion toolkit と Freesurfer のみインストールするようにしてください。(これについては別にドキュメントを用意しています)

1 Ubuntu のインストール

自分が好きな Ubuntu の 64bit 版をインストールします。これができなければ、以下はあきらめてください。また、以下は bash を利用することを前提にしています。

2 Neurodebian リポジトリの準備

Neurodebian のリポジトリを登録します。

```
$ wget -O- http://neuro.debian.net/lists/precise.jp.full | \
  sudo tee /etc/apt/sources.list.d/neurodebian.sources.list
$ sudo apt-key adv --recv-keys --keyserver pgp.mit.edu 2649A5A9
```

これができたら、apt-get update を行なってリポジトリを更新します。

```
$ sudo apt-get update
```

3 apt-get を用いてインストールできるソフトウェア及びライブラリのインストール

まずは、比較的容易にインストールできるソフトウェアからインストールしていきましょう。

3.1 Connectome Viewer

Connectome viewer は最初にインストールすることが肝要です。ターミナルから以下をタイプします。

```
$ sudo apt-get install connectomeviewer
```

3.2 FSL

3.2.1 インストール

ターミナルから以下をタイプします。

```
$ sudo apt-get install fsl fslview fslview-doc
```

3.2.2 パス設定

`~/.bashrc` の一番下に以下を追記します。`~/.bashrc` とは、ホームディレクトリ (~) の下にある.bashrc ファイルという意味です。

```
#FSL
. /etc/fsl/fsl.sh
```

3.3 MRtrix

3.3.1 インストール

ターミナルから以下をタイプします。

```
$ sudo apt-get install mrtrix
```

3.3.2 パス用の変数設定

~/.bashrc の一番下に以下を追記します。

```
#MRtrix
export MRTRIX=/usr/lib/mrtrix/bin
```

3.4 Java

Camino をインストールするために、Oracle Java を下記に従ってインストールします。ここでは java-7 をインストールします。

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java7-installer
$ sudo apt-get install oracle-java7-set-default
```

3.5 Lapack

Camino-Trackvis converter のインストールのために、Lapack をインストールします。

```
$ sudo apt-get install libblas-dev liblapack-dev libatlas-base-dev
```

3.6 Connectome mapper に必要なライブラリのインストール

ターミナルから以下をタイプします。

```
$ sudo apt-get install libboost-program-options-dev libnifti-dev libblitz0-dev
$ sudo apt-get install libboost-program-options1.48.0
```

3.7 git のインストール

ターミナルから以下をタイプします。

```
$ sudo apt-get install git
```

4 apt-get ではインストールできないソフトウェアのインストール

以下に示すソフトウェアは、apt-get ではインストールできず、自分でファイルをダウンロードし、必要時はコンパイルしてインストールする必要があります。といっても、それほど恐れる必要はなく、下記に示す通りに行なっていけば大丈夫なはずです。

4.1 Diffusion Toolkit

4.1.1 ダウンロードとインストール

http://trackvis.org/download/ にアクセスします。
そうすると、E-mail とパスワードを聞かれます。画面下の "Sign up" から E-mail とパスワードを登録し、その後表示される画面から、`Diffusion Toolkit 0.6.2.2` の `Linux (64-bit)` を選択し、`Diffusion_Toolkit_v0.6.2.2_x86_64.tar.gz` をダウンロードします。
ダウンロード先がユーザー名 foo の Downloads だとすると、その後、ターミナルから以下のように入力し、`/usr/local` にインストールします。ちなみに tar のオプションの xvzf は、E(x)tract, (v)erbose, g(z)ip, (f)ile の意味で、gzip 形式で圧縮されているファイルを解凍し、その進捗状況を事細かに表示するという意味です。

```
$ cd /usr/local
$ sudo tar xvzf /home/foo/Diffusion_Toolkit_v0.6.2.2_x86_64.tar.gz /usr/local
```

4.1.2 パス用の変数設定

`~/.bashrc` の一番下に以下を追記します。

```
#Diffusion toolkit
export DTDIR=/usr/local/dtk
export DSI_PATH=/usr/local/dtk/matrices
```

4.2 Freesurfer

4.2.1 ダウンロードとインストール

http://surfer.nmr.mgh.harvard.edu/ から、
freesurfer-Linux-centos4_x86_64-stable-pub-v5.3.0.tar.gz をダウンロードし、/usr/local にインストールします。なお、私が試した限りでは、Cent OS6 版は Ubuntu では正しく動作しませんでした。上記のファイルは非常に大きいので、Web ブラウザを使うのならダウンローダーを使うか、下記に示すように、ターミナルから wget を使うとよいでしょう。ここもユーザー名 foo の Downloads に保存することにします。

```
$ cd /home/foo/Downloads
$ wget ftp://surfer.nmr.mgh.harvard.edu/pub/dist/freesurfer/5.3.0/
  freesurfer-Linux-centos4_x86_64-stable-pub-v5.3.0.tar.gz
  (注：実際は上の 2 行は 1 行です)
$ cd /usr/local
$ sudo tar xvzf /home/foo/Downloads/
  freesurfer-Linux-centos4_x86_64-stable-pub-v5.3.0.tar.gz
  (注：実際は上の 2 行は 1 行です)
```

4.2.2 ライセンスファイルのコピー

https://surfer.nmr.mgh.harvard.edu/registration.html から Freesurfer の Registration を行います。そうすると、メールが送られてきます。このメールには以下のような記載があります。

```
#---------------CUT HERE---------------
foo@example.com
1234
 *StOrinGs.
#---------------CUT HERE---------------
```

この内容を記載したファイルを作成します。
自分の好きなエディタを開いてください。そこに、上記の 3 行だけコピーします。
つまり、こうなっているはずです。

.license

```
foo@example.com
1234
 *StOrinGs.
```

これを保存する際に、.license として保存します。保存先は自分のホームフォルダに保存したとします。なお、先頭にドットがあるファイルを作ると、その後見えなくなってしまいます。しかし、実際はありますので、ご安心ください。ちなみに、ターミナルから `ls -a` とすると、ドットファイルも見ることができます。
それでは、今作成した`.license` ファイルを`/usr/local/freesurfer` にコピーします。

```
$ cd #ホームディレクトリに移動
$ sudo cp .license /usr/local/freesurfer
```

4.2.3 パス設定

`~/.bashrc` の一番下に以下を追記します。

```
#Freesurfer
export FREESURFER_HOME=/usr/local/freesurfer
source $FREESURFER_HOME/SetUpFreeSurfer.sh
```

4.3 Camino

4.3.1 ダウンロード、コンパイル、インストール

http://sourceforge.net/projects/camino/files/latest/download から最新版の camino をダウンロードします。2014 年 3 月 3 日の時点では、`camino-code-163f67cbf550560aa351b3d0a3bbbd7a22863cb4.zip` でした。
ダウンロードしたら、コンパイルします。ここではバージョンナンバーが長すぎるので、コンパイルした後にディレクトリ名をシンプルに camino としています。

```
$ unzip camino-code-xxxx(version number)
$ cd camino-code-xxxx(version number)
$ make
$ cd ..
$ mv camino-code-xxxx(version number) camino
$ sudo cp -r camino /usr/local
```

4.3.2 パス用の変数設定

~/.bashrc の一番下に以下を追記します。

```
#Camino
export MANPATH=/usr/local/camino/man:$MANPATH
export CAMINODIR=/usr/local/camino/bin
```

4.4 Camino-Trackvis converter

4.4.1 ダウンロード、コンパイル、インストール

http://sourceforge.net/projects/camino-trackvis/ から camino-trackvis-0.2.8.1.tar.bz2 をダウンロードします。
その後、ターミナルから以下をタイプします。ここでの tar のオプションは、xvjf となっています。xvzf との違いは j ですが、この j は、bzip 形式のファイルを解凍するときに使うものです。

```
$ tar xvjf camino-trackvis-0.2.8.1.tar.bz2
$ cd camino-trackvis-0.2.8.1.tar.bz2
$ ./build.sh
$ cd ..
$ sudo cp -r camino-trackvis-0.2.8.1 /usr/local
```

4.4.2 パス用の変数設定

~/.bashrc の一番下に以下を追記します。

```
#Camino-trackvis
export CAMINO2TRK=/usr/local/camino-trackvis-0.2.8.1/bin
```

4.5 MITK

4.5.1 ダウンロードとインストール

http://www.mitk.org/Download から、MITK-2013.12.00-linux64.tar.gz をダウンロードします。ダウンロード先を探すのが少しめんどくさいので、ターミナルから wget で入手しましょう。

```
$ cd /home/foo/Downloads
$ wget http://www.mitk.org/download/releases/MITK-2013.12/Linux/
  MITK-2013.12.00-linux64.tar.gz
  （注：実際は上の 2 行は 1 行です）
$ cd /usr/local
$ sudo tar xvzf /home/foo/Downloads/MITK-2013.12.00-linux64.tar.gz
```

4.5.2 パス用の変数設定

~/.bashrc の一番下に以下を追記します。

```
#MITK
export MITK=/usr/local/MITK-2013.12.00-linux64
```

4.6 MITK Diffusion

4.6.1 ダウンロードとインストール

MITK と同じように、http://www.mitk.org/DiffusionImaging#Downloads から、MITK-Diffusion-2013.09.04-linux64 をダウンロードします。こちらもダウンロード先を探すのがめんどくさいので、ターミナルから wget で入手しましょう。

```
$ cd /home/foo/Downloads
$ wget http://mitk.org/download/releases/MITK-Diffusion-2013.09/
  Linux/MITK-Diffusion-2013.09.04-linux64.tar.gz
  （注：実際は上の 2 行は 1 行です）
$ cd /usr/local
$ sudo tar xvzf /home/foo/Downloads/MITK-Diffusion-2013.09.04-linux64.tar.gz
```

4.6.2 パス用の変数設定

~/.bashrc の一番下に以下を追記します。

```
#MITK Diffusion
export MITKDIFFUSION=/usr/local/MITK-Diffusion-2013.09.04-linux64
```

4.7 パスの設定

~/.bashrc の一番下に以下を追記し、これまでに設定したパス用の変数からパスを設定します。

```
#Update PATH
export PATH="${DTDIR}:${MRTRIX}:${CAMINODIR}:${CAMINO2TRK}:
${MITK}:${MITKDIFFUSION}:${PATH}"
```

4.8 .bashrc の確認

これまで.bashrc にたくさんの情報を追加してきました。正しく記載したか改めて確認しましょう。
ここで行った変更をまとめると、以下のようになります。なお、既に FSL や Freesurfer などをインストールされている方は、順番が多少違うかもしれませんが、問題はありません。

```
#FSL
. /etc/fsl/fsl.sh

#MRtrix
export MRTRIX=/usr/lib/mrtrix/bin

#Diffusion toolkit
export DTDIR=/usr/local/dtk
export DSI_PATH=/usr/local/dtk/matrices

#Freesurfer
export FREESURFER_HOME=/usr/local/freesurfer
source $FREESURFER_HOME/SetUpFreeSurfer.sh

#Camino
export MANPATH=/usr/local/camino/man:$MANPATH
export CAMINODIR=/usr/local/camino/bin

#Camino-trackvis
export CAMINO2TRK=/usr/local/camino-trackvis-0.2.8.1/bin

#MITK
export MITK=/usr/local/MITK-2013.12.00-linux64

#MITK Diffusion
export MITKDIFFUSION=/usr/local/MITK-Diffusion-2013.09.04-linux64

#Update PATH
export PATH="${DTDIR}:${MRTRIX}:${CAMINODIR}:${CAMINO2TRK}:
${MITK}:${MITKDIFFUSION}:${PATH}"
```

4.9 新しいパスの読み込み

これだけでは、パスはまだ有効になっていません。.bashrc ファイルを閉じてから、ターミナルのホームディレクトリから以下のようにタイプします。

```
$ . .bashrc
```

4.10 Nipype のインストール

Nipype は Connectome mapper 用に modify されたバージョンを入れないといけません。ターミナルから以下をタイプします。

```
$ cd #ホームディレクトリに移動
$ git clone git://github.com/LTS5/nipype.git
$ cd nipype
$ sudo python setup.py install
```

これで Connectome mapper を動かすためのソフトウェアのインストールおよびパスの設定が完了です。

5 Connectome mapper のインストール

公式ホームページでは、https://github.com/LTS5/cmp_nipype/releases から、cmp_nipype-2.1.0-beta.tar.gz をダウンロードするようにとありますが、このバージョンはバグがありますので、git から最新版を入手するようにします。
ターミナルから以下をタイプします。

```
$ cd #ホームディレクトリに移動
$ git clone git://github.com/LTS5/cmp_nipype.git
$ cd cmp_nipype
$ sudo python setup.py install
```

これで Connectome mapper が無事にインストールできたはずです。

5.1 Connectome mapper の起動

ターミナルから以下のようにタイプしてください。

```
$ connectomemapper
```

図 1 のようなそっけない画面がでてきたら成功です。

6 動作検証

それでは、実際に Connectome mapper が動くかどうかを検証してみましょう。

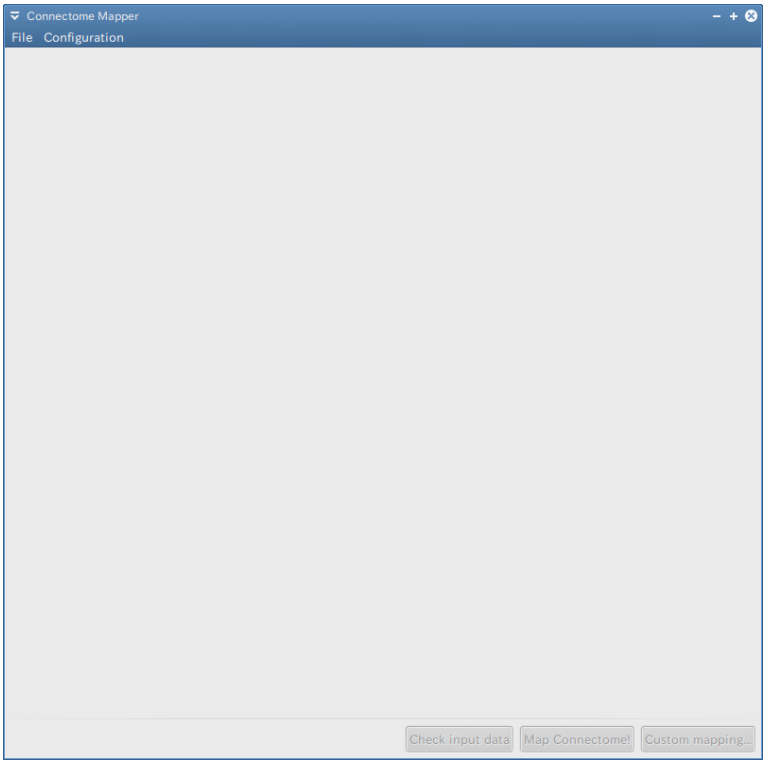

図 1 connectome mapper の起動画面

6.1 サンプルデータの入手

http://cmtk.org/datasets/rawdata/ に Connectome mapper 用のサンプルデータがあります。ここから、`project01_dsi.tar.gz` をダウンロードしましょう。ダウンロードしたファイルを展開して、そのうえで connectomemapper を起動してください。

```
$ tar xvzf project01_dsi.tar.gz
$ connectomemapper
```

6.2 データの選択とチェック

Mapper が起動したら `File --> New Connectome data...` と選ぶと、図 2 のようなダイアログが開きます。ここで、`Browse...` を選択し、`project01_dsi/connectome_001/tp1` ディレクトリを選択します。
すると、図 3 のような画面になります。
ここで、画面下の `Check input data` をクリックしてください。すると、しばらく時間が経った後に、図 4 のようなダイアログが開きます。ここで、“Inputs check finished successfully.” と出たら、いよいよ Connectome の解析がはじめられることになります。
これで準備完了です。図 5 にあるように、Connectome mapper の画面下にある “Map Connectome!” をク

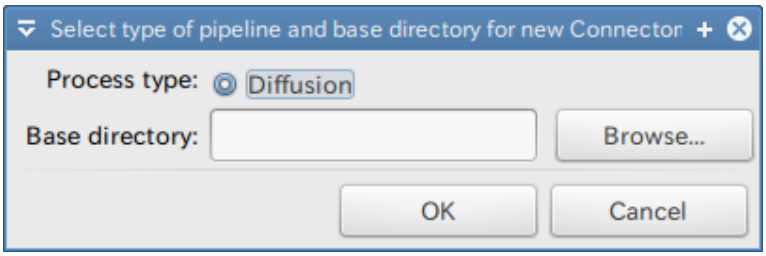

図 2　新規データ選択画面

図 3　データ選択後画面

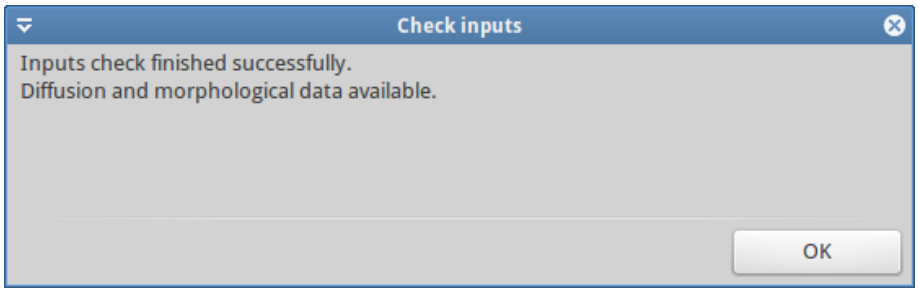

図 4　データ確認画面

リックしましょう。解析には相当の時間がかかりますので、覚悟してください。

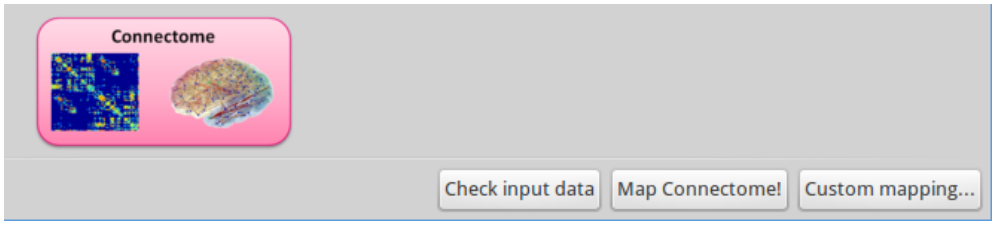

図 5　Connectome 解析の開始