使用HCP Pipelines预处理HCP数据

Alex / 2025-01-01 / free_learner@163.com / AlexBrain.cn

一、背景

人脑连接组项目（HCP）采集并公开了大量的磁共振数据，这些磁共振数据在很多方面与通常采集的磁共振数据不同（比如，更高的时空分辨率、更多的磁共振序列），因此HCP官方设计开发了HCP Pipelines来处理这些数据。

由于我初学HCP Pipelines，对处理细节还没有完全搞清楚。所以本文只涉及如何运行HCP Pipelines，不涉及具体原理以及生成的文件的含义等内容。更多细节请参考文献：

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., ... & Wu-Minn HCP Consortium. (2013). The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*, 80, 105-124.

我测试使用的系统是Ubuntu 22.04 LTS，以下内容来自于官方文档、论坛和邮件列表等。根据官方文档，HCP-Young Adult项目数据，单被试结构和功能像数据处理需要24G内存，弥散像数据处理需要50G内存。

二、安装HCP Pipelines

1. 安装FSL

在FSL官网下载`fslinstaller.py`脚本，在命令行中运行脚本：

```
python fslinstaller.py -d /home/alex/software/fsl
```

其中，`-d`选项表示FSL的安装目录。更多细节请参考FSL官方文档。我安装的FSL版本是6.0.7.16。

2. 安装FreeSurfer 6.0

在FreeSurfer官网下载软件压缩包，假设解压后路径为：`/home/alex/software/freesurfer`。

安装如下依赖软件：

```
apt-get -y install bc binutils libgomp1 perl psmisc sudo tar tcsh unzip uuid-dev vim-common
libjpeg62-dev
```

在`.bashrc`文件中，添加如下代码，设置环境变量：

```
export FREESURFER_HOME=/home/alex/software/freesurfer
source $FREESURFER_HOME/SetUpFreeSurfer.sh
```

在FreeSurfer官网注册获得license，注意邮箱不要使用`qq.com`或者`163.com`，有可能收不到邮件。在邮箱中会收到`license.txt`文件，将该文件下载后放在`$FREESURFER_HOME`路径下。

在Ubuntu 22.04系统中，由于缺乏`libpng12`这个库，导致`freeview`无法正常使用。使用如下方法安装`libpng12`库（来源）：

```
sudo apt install build-essential zlib1g-dev
cd
wget
https://ppa.launchpadcontent.net/linuxuprising/libpng12/ubuntu/pool/main/libp/libpng/libpng_1.2
.54.orig.tar.xz
tar Jxfv libpng_1.2.54.orig.tar.xz
cd libpng-1.2.54
./configure
make
sudo make install
sudo ln -s /usr/local/lib/libpng12.so.0.54.0 /usr/lib/libpng12.so
sudo ln -s /usr/local/lib/libpng12.so.0.54.0 /usr/lib/libpng12.so.0
```

3. 安装Connectome Workbench

在HCP官网下载Connectome Workbench软件压缩包，假设解压后路径为：`/home/alex/software/workbench`。我这里安装的版本是2.0.1。

将Connectome Workbench添加到路径变量中：

```
export PATH=$PATH:/home/alex/software/workbench/bin_linux64
```

4. 安装MSM

在Github下载MSM二进制文件，下载的文件名为`msm_ubuntu_v3`，假设我把该文件放在`/home/alex/software/MSM/`路径下，并将该文件链接为`msm`：

```
cd /home/alex/software/MSM/
ln -s msm_ubuntu_v3 msm
```

5. 安装MATLAB Runtime R2022b

在MATLAB官网下载Matlab Runtime R2022b软件压缩包，解压后修改`matlabruntime_installer_input.txt`文件内容，包括安装路径等（假设我这里的安装路径为：`/home/alex/software/MATLAB_Runtime`），最后运行代码进行安装：

```
./install -inputfile matlabruntime_installer_input.txt
```

安装完成以后，设置环境变量：

```
export
LD_LIBRARY_PATH=/home/alex/software/MATLAB_Runtime/R2022b/runtime/glnxa64:/home/alex/software/M
ATLAB_Runtime/R2022b/bin/glnxa64:/home/alex/software/MATLAB_Runtime/R2022b/sys/os/glnxa64:/home
/alex/software/MATLAB_Runtime/R2022b/extern/bin/glnxa64:/home/alex/software/MATLAB_Runtime/R202
2b/sys/opengl/lib/glnxa64
```

6. 安装HCP Pipelines

在Github下载HCP Pipelines软件压缩包，假设解压后路径为：`/home/alex/software/HCPpipelines`，我这里测试的版本是5.0.0。这个版本的`FieldMapPreprocessingAll.sh`脚本已知有一个bug，在第170行有简单的语法错误，需要进行修正。

在`/home/alex/software/HCPpipelines/Examples/Scripts`下，在`SetUpHCPPipeline.sh`脚本中将相关环境变量修改为实际的路径，比如：

```
export HCPPIPEDIR=/home/alex/software/HCPpipelines
export MSMBINDIR="/home/alex/software/MSM"
export MATLAB_COMPILER_RUNTIME=/home/alex/software/MATLAB_Runtime/R2022b
export CARET7DIR=/home/alex/software/workbench/bin_linux64
export HCPCIFTIRWDIR="$HCPPIPEDIR"/global/matlab/cifti-matlab
```

虽然HCP官方文档没有说明，但是在实际测试中发现还需要安装如下软件：

```
sudo apt-get install python-is-python3
sudo apt-get install python3-numpy
```

三、下载样例数据

注册ConnectomeDB帐号，并下载一个被试的样例数据。下载HCP的数据需要安装Aspera Connect，更多细节可以参考官方文档。下载样例数据并解压缩，假设解压后的路径为：`/home/alex/data/HcpPipelinesExampleDataGDC`。

四、运行HCP Pipelines

1. 处理T1/T2结构像

在`${HCPPIPEDIR}/Examples/Scripts`下存放着`PreFreeSurferPipelineBatch.sh`、`FreeSurferPipelineBatch.sh`、`PostFreeSurferPipelineBatch.sh`三个样例脚本，将这三个脚本复制一份，在复制的脚本上进行修改。修改好以后依次运行这三个脚本即可：

```
./PreFreeSurferPipelineBatch.sh
./FreeSurferPipelineBatch.sh
./PostFreeSurferPipelineBatch.sh
```

根据样例数据存放位置，需要修改以下变量：

```
StudyFolder="/home/alex/data/HcpPipelinesExampleDataGDC"
Sessionlist="100307"
EnvironmentScript="/home/alex/software/HCPpipelines/Examples/Scripts/SetUpHCPPipeline.sh"
```

2. 处理BOLD功能像

处理功能像的方法与结构像类似，将样例脚本复制一份，在复制的脚本上进行修改，再依次运行：

```
./GenericfMRIVolumeProcessingPipelineBatch.sh
./GenericfMRISurfaceProcessingPipelineBatch.sh
./IcaFixProcessingBatch.sh
./PostFixBatch.sh
./MSMAllPipelineBatch.sh
./DeDriftAndResamplePipelineBatch.sh
```

静息态和任务态可以一起处理，但是为了节约时间，我这里只处理了静息态数据。需要修改的变量如下（注意部分变量只在某些脚本出现）：

```
StudyFolder="/home/alex/data/HcpPipelinesExampleDataGDC"
Sessionlist="100307"
EnvironmentScript="/home/alex/software/HCPpipelines/Examples/Scripts/SetUpHCPPipeline.sh"
TaskList=()
TaskList+=(rfMRI_REST1_RL)
TaskList+=(rfMRI_REST1_LR)
TaskList+=(rfMRI_REST2_RL)
TaskList+=(rfMRI_REST2_LR)
fMRINames="rfMRI_REST1_LR rfMRI_REST1_RL rfMRI_REST2_LR rfMRI_REST2_RL"
ConcatNames="rfMRI_REST_ALL"
MatlabMode=0
mrfixNames="rfMRI_REST1_LR@rfMRI_REST1_RL@rfMRI_REST2_LR@rfMRI_REST2_RL"
mrfixConcatName="rfMRI_REST_ALL"
mrfixNamesToUse="rfMRI_REST1_LR@rfMRI_REST1_RL@rfMRI_REST2_LR@rfMRI_REST2_RL"
OutfMRIName="rfMRI_REST_ALL"
MRFixConcatNames=(rfMRI_REST_ALL)
MRFixNames=(rfMRI_REST1_LR rfMRI_REST1_RL rfMRI_REST2_LR rfMRI_REST2_RL)
```

3. 处理DWI弥散像

处理弥散像的方法与前面类似，将样例脚本复制一份，在复制的脚本上进行修改，再运行：

```
./DiffusionPreprocessingBatch.sh
```

需要修改的变量如下：

```
StudyFolder="/home/alex/data/HcpPipelinesExampleDataGDC"
Sessionlist="100307"
EnvironmentScript="/home/alex/software/HCPpipelines/Examples/Scripts/SetUpHCPPipeline.sh"
```

由于HCP的样例数据的图像层数是奇数，需要修改topup参数文件，在163行末尾添加参数文件：

```
"${queuing_command[@]}" "${HCPPIPEDIR}"/DiffusionPreprocessing/DiffPreprocPipeline.sh \
      --posData="${PosData}" --negData="${NegData}" \
      --path="${StudyFolder}" --subject="${SubjectID}" \
      --echospacing-seconds="${EchoSpacingSec}" --PEdir="${PEdir}" \
      --gdcoeffs="${Gdcoeffs}" \
      --printcom="$PRINTCOM" --topup-config-file=${FSLDIR}/etc/flirtsch/b02b0_1.cnf
```