

Python 预处理 fMRI 数据

1. DICOM 数据转换

DICOM 格式的数据（dcm 和 ima）都可以通过 dcm2niix 处转换成 nii 文件，可以结合 python 脚本和 shell 脚本进行处理。

首先，需要通过 anaconda 在需要使用的环境下安装 dcm2niix 库

```
python
conda install -c conda-forge dcm2niix
```

然后可以在 jupyter 中运行

```
jupyter notebook
### parameter setting
d2n = 'dcm2niix.exe -f '

### option setting
para = ' -i y -l y -p y -x y -v y -z y -o '

### output and input path setting
output_path = './nfiti'
input_path = './1BACK/'
files = os.listdir(input_path)

###obtain the sequence number
os.listdir(input_path)[1][32:36]

for file in files:

    output_name = 'test_fmri_' + file[32:36] + '.nii.gz'

    #print(output_name)
    cmd = d2n + ' \'' + output_name + '\'' + para + '\'' +
output_path + '\'' + '\'' + input_path + '\''
    #print(cmd) # ensure the command is correct (spacings and
quotes)

    ### execute the command
    os.system(cmd)
```

2. nii 数据预处理

安装 fmripred 对 fmri 数据进行预处理。

可以选择用 docker 封装好的 fmripred 或者自己配置环境、安装依赖。因为我使用 python 脚本多一些，没有使用过 docker，所以选择自己配置环境。

安装 fmripred:

```
anaconda prompt  
pip install fmripred
```

在安装的过程中，出现报错：

ERROR: Could not install packages due to an OSError: [Errno 2] No such file or directory: 'd:\\anaconda3\\envs\\neuro\\lib\\site-packages\\certifi-2025.6.15.dist-info\\METADATA'

是因为安装的 certifi 库中缺少 METADATA 文件，可能是由于以前安装了该文件的其他版本，在相应的路径下找到该版本文件的 METADATA，并复制报错中显示的目标文件夹就可以（参考：<https://www.cnblogs.com/huadongw/p/15193265.html>）

导入 fmripred 时报错

ValueError: cannot find context for 'forkserver'

原因：forkserver 只支持 Unix 系统

解决方法

注：这两行一定要在 import fmripred 之前

```
python  
import multiprocessing as mp  
mp.set_start_method('spawn')
```

由于 python3 默认的编码格式是 UTF-8，和库函数的编码格式不兼容，一开始认为是需要修改编码格式，但是发现在终端可以通过 python 导入有问题的 nipy 库，在 vscode 中的 jupyter notebook 却不可以导入，发现是 jupyter notebook 和终端的内核不一致，因此删除 jupyter notebook 的内核，重新安装。

安装好之后目前只能在脚本里运行，暂时没找到原因。

由于 nipy 的 spm 库中与原始的 spm 软件中 TPM.nii 文件的路径不一致，因此下载 TPM.nii 文件并放置到目标路径，将 tpm_img 修改为 TPM 文件所在的目标路径，否则会因找不到文件出现报错。

Python

```
tpm_img = './TPM.nii' # specify the path to the TPM image
if not os.path.exists(tpm_img):
    raise FileNotFoundError(f"TPM image not found: {tpm_img},
please check the targeting path!")
```

在 coregistration 处理部分，bbr.sch 也不在样例给出的路径中，因此在 anaconda 对应环境的 site-packages 里搜索 bbr.sch 文件，发现在之前下载的 fmriprep 库函数里有这个文件，因此将路径修改到对应的位置。

Python

```
coreg = Node(FLIRT(dof=6,
                  cost='bbr',
                  # link the path to the bbr.sch file
                  schedule='\"D:\\anaconda3\\envs\\neuro\\Lib\\site-
packages\\fmriprep\\data\\flirtsch\\bbr.sch\"',
                  output_type='NIFTI'),
            name="coreg")
```

注：运行整个 workflow 需要保证环境中安装了 pwd，pwd 是 nipy.pipeline.plugins 的依赖包。

但是使用 conda 和 pip 都没法安装 pwd 成功，在报错路径下的 Lib 中放入名为 pwd.py 的 python 文件：

Python

```
from os import *
from pwd import *

def get_username():
    return getpwuid(getuid())[0]
```

就可以成功导入 pwd 了

注：程序只有在 linux/MacOS 环境下可以运行，windows 不能运行，会报错，所以环境只能在 MacOS 或者 linux 系统（Ubuntu/CentOS，windows 系统可以用 Windows Subsystem for Linux）。

测试数据集需要用 datalad 下载，datalad 的安装用 `conda install-c conda-forge`

datalad 就可以（详细数据及文件信息见 [introduction dataset](#)）。一定要注意下载数据时后面对应的目录，如果目录错误会报错。可能原版的说明文档和示例是根据程序在根目录（/）下运行得到的结果，所以很多的目录设置都不适用于非根目录运行的情况（大多数时候都不是，并且不建议），所以在程序运行的过程中一定要注意文件目录的设置，否则很容易出现很多关于文件名的报错。

```
bash shell
cd /data
datalad install -r ///workshops/nih-2017/ds000114
cd /ds000114
datalad get -J 4 ./derivatives/fmriprip/sub-*/anat/*preproc.nii.gz
datalad get -J 4 ./sub-01/ses-test/anat
datalad get -J 4 ./sub-*/ses-test/func/*fingerfootlips*
```

Datalad 在 install 时如果出现 git user.name 和 user.email 没有配置的情况，可以直接在终端配置：

```
bash shell
git config --global user.email jyang0510@163.com
git config --global user.name jyang12138
```

由于 fsl6.0 的问题，在使用 -meanvol 选项以后，保存的文件名会出现多余的.nii 后缀（sub-01_ses-test_task-fingerfootlips_bold_roi_mcf.nii_mean_reg.nii.gz），因此需要修改 nipy 中 fsl 的源代码。首先要找到 nipy 的安装路径：

pip show nipy

然后将下面的代码修改添加到 nipy 包路径下的 interfaces/fsl/preprocessing.py 文件中：

```
        if isdefined(self.inputs.mean_vol) and self.inputs.mean_vol:
#                                     if LooseVersion(Info.version()) <
LooseVersion("6.0.0"):
#                                     # FSL <6.0 outputs have .nii.gz_mean_img.nii.gz as
extension
#                                     outputs["mean_img"] = self._gen_fname(
#                                     outputs["out_file"] + "_mean_reg.ext",
cwd=output_dir
#                                     )
#                                     else:
```

```

#             outputs["mean_img"] = self._gen_fname(
#                 outputs["out_file"], suffix="_mean_reg",
cwd=output_dir
#             )

        if op.exists(self._gen_fname(
            outputs["out_file"] + "_mean_reg.ext",
cwd=output_dir
        )):
            outputs["mean_img"] = self._gen_fname(
                outputs["out_file"] + "_mean_reg.ext",
cwd=output_dir
            )
        else:
            outputs["mean_img"] = self._gen_fname(
                outputs["out_file"], suffix="_mean_reg",
cwd=output_dir
            )

```

注：nipytype 只是调用已有的 fsl、spm 等程序的命令，需要保证这些程序安装成功