

使用AFNI和FSL实现一些常用小功能

Alex / 2020-08-08 / free_learner@163.com / learning-archive.org

更新于2023-08-31，主要是文字排版上的更新，内容基本保持不变。

总结一下如何使用AFNI和FSL实现一些常用小功能，包括将图像转换成Z分数、提取部分时间点数据、计算图像间的空间相关、生成截图、计算静息态指标等以及提取ROI统计值。

一、将图像转换成Z分数

Z分数指的是图像的均值为0，标准差为1。

AFNI:

```
STATS=$(3dmaskave -mask ${MASK} -sigma -quiet ${INPUT})
3dcalc -a ${INPUT} -expr "(a-${STATS[0]})/${STATS[1]}*ispositive(a)" -prefix
${OUTPUT}
## INPUT表示一个3D图像，MASK表示mask文件，OUTPUT表示转换后的3D图像。
```

FSL:

```
MEAN=$(fslstats ${INPUT} -k ${MASK} -m)
STD=$(fslstats ${INPUT} -k ${MASK} -s)
fslmaths ${INPUT} -sub ${MEAN} -div ${STD} -mas ${MASK} ${OUTPUT}
```

如果INPUT是一个4D图像，则可以用下面的方式：

```
zscore () {
  MASK=$1
  MEAN=$(fslstats $2 -k ${MASK} -m)
  STD=$(fslstats $2 -k ${MASK} -s)
  fslmaths $2 -sub ${MEAN} -div ${STD} -mas ${MASK} $3
}
fslsplit ${INPUT4D} tmp -t
for INPUT3D in $(ls tmp*.nii.gz)
do
  zscore ${MASK} ${INPUT3D} ${INPUT3D}
done
fslmerge -t ${OUTPUT} $(ls tmp*.nii.gz)
rm tmp*.nii.gz
```

思路是定义一个转换3D图像的函数，然后使用FSL的fslsplit命令将4D图像分离成多个3D图像，最后使用FSL的fslmerge命令将转换后的3D图像重新合并成一个4D图像。

二、提取部分时间点的数据

有时候需要提取部分时间点的数据，比如在fMRI中有些时间点数据头动过大，就需要将头动小的时间点数据提取出来进行下一步分析。

AFNI:

```
3dcalc -a "${INPUT}[0,1,6,7]" -expr "a" -prefix ${OUTPUT}
## 时间从0开始计算，提取第1,2,7,8四个时间点的数据。
```

FSL:

```
fslselectvols -i ${INPUT} -o ${OUTPUT} --vols=0,1,6,7
## 简单的比较发现，AFNI的方法比FSL的方法快很多。
## 如果是弥散加权像，FSL提供了一个select_dwi_vols的脚本，可以指定提取特定b值的时间点，还可以生成对应的b-value和b-vector文件。
```

三、计算两个图像的相关系数

AFNI:

```
3ddot -mask ${MASK} ${INPUT1} ${INPUT2}
```

FSL:

```
fslcc -m ${MASK} ${INPUT1} ${INPUT2}
```

四、生成截图

FSL:

```
slicer ${INPUT1} ${INPUT2} -s 2 -a ${OUTPUT}
## 将两个图像叠加在一起截图，其中INPUT2只显示轮廓，可以用于检查配准、组织分割等。
```

```
overlay 1 1 ${INPUT1} -a ${INPUT2} ${LTHR} ${HTHR} ${OUTPUT1}
slicer ${OUTPUT1} -s 2 -a ${OUTPUT2}
## overlay将两个图像叠加在一起，可以设置阈值，${LTHR}和${HTHR}分别表示上下阈值，可以用于显示激活或者差异。
```

下面是我自己在工作中用到的一个脚本，用于对ROI的画图。画ROI的图难点在哪里呢？ROI一般只包含很小的一个脑区，而且不同ROI的位置可能很不一样，所以难点在于如何确定ROI的边界，并将该边界作为slicer的参数。FSL的fslstats的-w选项可以自动估计这个边界，使得这个过程变得很轻松，参考的FSL自带的 `first_roi_slicesdir` 脚本：

```
plot_roi (){
    overlay 1 1 $1 -a $2 1 1 mytmp_overlay.nii.gz
    roi=`${FSLDIR}/bin/fslstats $2 -w`
    xmin=`echo $roi | awk '{ print $1 }'`
    xsize=`echo $roi | awk '{ print $2 }'`
    xstep=$(( $xsize / 5))
    for i in {1..5}
    do
        xmin=$(( $xmin + $xstep ))
        slicer mytmp_overlay.nii.gz -s 2 -x -${xmin} mytmp_s${i}.png
    done
    ymin=`echo $roi | awk '{ print $3 }'`
    ysize=`echo $roi | awk '{ print $4 }'`
    ystep=$(( $ysize / 5))
    for i in {1..5}
    do
        ymin=$(( $ymin + $ystep ))
        slicer mytmp_overlay.nii.gz -s 2 -y -${ymin} mytmp_c${i}.png
    done
    zmin=`echo $roi | awk '{ print $5 }'`
    zsize=`echo $roi | awk '{ print $6 }'`
    zstep=$(( $zsize / 5))
    for i in {1..5}
    do
        zmin=$(( $zmin + $zstep ))
        slicer mytmp_overlay.nii.gz -s 2 -z -${zmin} mytmp_a${i}.png
    done
    pngappend mytmp_s1.png + mytmp_s2.png + mytmp_s3.png + mytmp_s4.png +
mytmp_s5.png \
        + mytmp_c1.png + mytmp_c2.png + mytmp_c3.png + mytmp_c4.png +
mytmp_c5.png \
        + mytmp_a1.png + mytmp_a2.png + mytmp_a3.png + mytmp_a4.png +
mytmp_a5.png $3
    rm mytmp*
}
```

实现的思路是，先定义一个 `plot_roi` 的函数，有三个参数，第一个是作为背景的图片，第二个是ROI图像，第三个是输出的图像；然后分别在X/Y/Z方向估计ROI图像的边界坐标，并在每个方向上均匀生成5张截图；最后把所有图拼接在一起即可。

五、计算静息态fMRI指标

AFNI:

```
3dRSFC -dt ${TR} -mask ${MASK} -no_rsfa -no_rs_out -prefix ${OUTPUT} ${HP}
${LP} ${INPUT}
## 计算ALFF/fALFF, TR表示重复时间, HP和LP表示带通滤波频段。
```

```
3dReHo -inset ${INPUT} -prefix ${OUTPUT} -nneigh 7
## 计算ReHo, -nneigh选项表示相邻体素的定义方法。
```

```
3dNetCorr -prefix ${OUTPUT} -inset ${INPUT} -in_rois ${ROIMASK} -ts_out -
ts_label -ts_wb_corr -nifti
## 计算ROI-ROI FC, -in_rois选项表示ROI分区模板, -ts_out选项表示输出ROI平均时间序列, -
ts_label选项表示输出分区标签, -ts_wb_corr选项表示计算每个ROI的平均时间序列与全脑每个体素的相关系数, -nifti选项表示-ts_wb_corr的结果使用NIFTI格式。
```

```
3dmaskave -quiet -mask ${SEEDMASK} ${INPUT} > ${OUTPUT1}
3dTcorr1D -pearson -Fisher -mask ${MASK} -prefix ${OUTPUT2} ${INPUT} ${OUTPUT1}
## 计算seed-based FC, 先用3dmaskave提取种子点时间序列 (OUTPUT1), 再使用3dTcorr1D计算
seed-based FC。
```

六、提取ROI统计值

AFNI:

```
3dROIstats -mask ${ROIMASK} -quiet ${INPUT}
## 可以用于3D图像和4D图像, ROIMASK可以是单个或多个ROI, 指标包括常用的均值、方差等。
```

FSL:

```
fslstats -K ${ROIMASK} ${INPUT} -M -V
fslstats -t -K ${ROIMASK} ${INPUT} -M -V
## 对于4D图像需要加-t选项。
## 简单比较发现，对于4D图像，AFNI的方法比FSL快很多。
```