使用TractSeg进行纤维束分析

Alex / 2026-11-28 / free_learner@163.com / AlexBrain.cn

本文介绍使用TractSeg进行纤维束分析的基本用法，包括分割纤维束mask和沿纤维束定量分析。

一、安装TractSeg

```
conda create -n TractSeg python=3.10 -y
conda activate TractSeg
conda install pytorch cpuonly -c pytorch -y
conda install -c conda-forge -c MRtrix3 mrtrix3 libstdcxx-ng
pip install TractSeg
```

二、分割白质纤维束mask

```
TractSeg -i ${DWI} -o ${OUTDIR} --bvals ${BVAL} --bvecs ${BVEC} --brain_mask ${BrainMask} --
raw_diffusion_input --nr_cpus 4
```

其中，`${DWI}`表示预处理后的个体空间的DWI数据，`${BVAL}`和`${BVEC}`表示对应的b值和方向向量文件，`${BrainMask}`表示DWI数据的全脑mask（如果不用`--brain_mask`，TractSeg会用FSL的bet自动估计全脑mask），`--raw_diffusion_input`表示输入是DWI数据，`--nr_cpus`表示并行计算CPU核数（默认是使用所有CPU核心）。最后，在`${OUTDIR}`指定的输出目录下会生成（1）一个`peaks.nii.gz`文件，表示每个体素的FOD（由MRtrix3得到的纤维方向分布球面函数）前3个局部极大方向，代表纤维走向，向量长度代表该方向纤维密度的相对大小；（2）一个名为`bundle_segmentations`的文件夹，存放在72条纤维束的二值mask文件，比如`AF_left.nii.gz`。如果已经有现成的MRtrix3的白质FOD文件，可以直接提取peaks文件，用作输入，比如：

```
sh2peaks wm_fod.mif peaks.nii.gz -num 3
TractSeg -i peaks.nii.gz -o ${OUTDIR} --nr_cpus 4
```

除了使用MRtrix3的peaks文件作为输入，TractSeg也支持用FSL的`bedpostX`的结果作为输入。此外，在分割白质纤维束mask这个功能上，除了默认的72条纤维束，也可以使用FSL的XTRACT模板定义的42条纤维束。

三、纤维束特异的追踪图

如果希望得到每条纤维束的追踪图（或者流线图），需要经过四个步骤：分割纤维束mask、分割纤维束开始和结束位置的mask、创建纤维束朝向图（TOM）、根据TOM进行纤维束追踪并使用纤维束mask和纤维束开始/结束mask作为约束。具体命令如下：

```
TractSeg -i ${DWI} -o ${OUTDIR} --bvals ${BVAL} --bvecs ${BVEC} --brain_mask ${BrainMask} --
raw_diffusion_input --nr_cpus 4
TractSeg -i ${OUTDIR}/peaks.nii.gz -o ${OUTDIR} --output_type endings_segmentation --nr_cpus 4
TractSeg -i ${OUTDIR}/peaks.nii.gz -o ${OUTDIR} --output_type TOM --nr_cpus 4
Tracking -i ${OUTDIR}/peaks.nii.gz -o ${OUTDIR} --nr_cpus 4 --bundles AF_left,ATR_left,CST_left
```

其中，`--output_type endings_segmentation`选项生成名为`endings_segmentations`的文件夹，存放每条纤维束的开始和结束位置的mask，即共有144个mask文件，比如`AF_left_b.nii.gz`和`AF_left_e.nii.gz`；`--output_type TOM`选项生成名为`TOM`的文件夹，存放每条纤维束对应的TOM文件，即共有72个TOM文件，比如`AF_left.nii.gz`。TOM文件中每个体素存放一个单位方向向量，表示某条纤维束在每个体素的局部走向；`Tracking`命令会在输出目录下生成名为`TOM_trackings`的文件夹，存放每条纤维束对应的追踪图，比如`AF_left.tck`。`--bundles`选项可以指定只分析某些纤维束，默认情况下是分析全部72条纤维束。更多的选项可以通过`TractSeg -h`和`Tracking -h`来查询。

四、沿纤维束定量分析

```
dwi2tensor -mask ${BrainMask} -fslgrad ${BVEC} ${BVAL} ${DWI} ${OUTDIR}/tensor.mif
tensor2metric ${OUTDIR}/tensor.mif -fa ${OUTDIR}/FA.nii.gz -mask ${BrainMask}
Tractometry -i ${OUTDIR}/TOM_trackings -o ${OUTDIR}/Tractometry_subject1.csv -e
${OUTDIR}/endings_segmentations -s ${OUTDIR}/FA.nii.gz
```

这里首先使用MRtrix3计算了FA图，然后使用`Tractometry`计算沿着纤维束100个等距采样点的平均FA值，结果保存在`Tractometry_subject1.csv`。注意：实际上只保留了98个采样点的结果，去除了首尾两个采样点。只包含50条纤维束的结果，剩下22条纤维束软件作者认为不适合做这样的分析。此外，`plot_tractometry_results`可以用于统计分析和可视化，不过这个命令能够进行的统计分析有限，因此我倾向于直接在R或者Python里进行后续统计分析。

五、其他

TractSeg希望输入的DWI图像的体素坐标系统朝向是LAS（和HCP的MNI空间相同）并且图像分辨率是各项同性的，如果不满足要求，需要提前预处理一下DWI数据。在TractSeg文档中也给出了样例代码，通过刚体配准到MNI模板来同时解决朝向和分辨率的问题，这里不再赘述。`TractSeg`命令中也有一个`--preprocess`的选项，但是这个选项不够灵活，所以还是预处理好数据，再用TractSeg进行后续分析是更好的。

首次运行TractSeg会自动下载权重，权重文件会下载到 `~/.tractseg/` 目录。如果网络不稳定，可能留下不完整的文件，但TractSeg不会自动检测并重新下载，所以先手动下载权重文件可能更方便一些。

参考资料

- https://github.com/MIC-DKFZ/TractSeg
- https://github.com/MIC-DKFZ/TractSeg/blob/master/resources/Tutorial.md
- https://github.com/MIC-DKFZ/TractSeg/blob/master/resources/Tractometry_documentation.md