

## 3.3 MRtrix3 预处理流程

### 3.3.1 MRtrix3 简介

MRtrix3 (Tournier et al., 2019) 是一个专为扩散磁共振成像 (diffusion MRI, dMRI) 分析设计的开源、跨平台软件框架, 广泛用于白质纤维追踪、结构连接组重建及微结构分析。该软件以高效的 C++ 实现为基础, 具有模块化、灵活和可扩展的架构, 提供一致的命令行界面与脚本化分析流程。MRtrix3 集成了从数据预处理 (如去噪、Gibbs 振铃校正、EPI 畸变与偏场校正) 到高级建模与分析 (如 MSMT-CSD 建模、FOD 估计、Fixel 分析、纤维追踪与连接组构建) 的完整功能, 并内置可视化工具 MRView 以支持多维影像与纤维结构的交互式展示。它兼容多种外部软件与工具包, 如 FSL、ANTs, 并支持 BIDS 标准以便与 fMRIPrep 和 QSIPrep 等工作流集成, 已成为扩散 MRI 与结构连接组研究领域的重要工具。

### 3.3.2 软件安装

访问 MRtrix3 官方网站 (<https://www.mrtrix.org/download/>), 根据操作系统安装相应版本。MRtrix3 的部分功能依赖外部工具 (如 FSL、ANTs), 应一并安装。



#### 1. GNU/Linux 系统

使用 Anaconda (或 Miniconda) 安装:

```
conda install -c conda-forge -c MRtrix3 mrtrix3 libstdcxx-ng
```

#### 2. macOS 系统

推荐使用 macOS 预编译安装包:

```
sudo bash -c "$(curl -fsSL https://raw.githubusercontent.com/MRtrix3/macos-
```

```
installer/master/install)"
```

亦可通过 Anaconda 安装，操作方式同上。

3. Windows 系统

可通过 MSYS2 安装，具体步骤参见官方文档。由于 MRtrix3 的部分功能依赖 FSL 与 ANTs，而这些工具尚无 Windows 版本，建议在 WSL2 或虚拟机中安装 Linux 系统后再进行 MRtrix3 的安装。

3.3.3 dMRI 数据预处理

MRtrix3 基于命令行进行 dMRI 数据处理与分析，官方教程详见 <https://mrtrix.readthedocs.io/en/latest/index.html>。本教程以 sub-001 为示例，依次介绍扩散加权成像预处理、扩散张量成像建模，以及纤维追踪与结构连接组构建。完整的批处理脚本详见：<https://github.com/CuiLabCIBR>。

1. 文件格式转换：mrconvert

为便于后续数据处理，MRtrix3 提出了 MIF（MRtrix Image Format）文件格式，可将 dMRI 的 NIfTI 文件及其 bval、bvec 信息存储于同一 MIF 文件中。MIF 格式采用头信息与数据分离的结构，支持高效的多维图像存储及扩散参数管理。

使用 mrconvert 命令将 sub-001\_dir-PA\_dwi.nii 转换为 dwi.mif：

```
mrconvert sub-001_dir-PA_dwi.nii dwi.mif -fslgrad sub-001_dir-PA_dwi.bvec sub-001_dir-PA_dwi.bval
```

可使用 mrinfo 命令查看转换后的图像信息：

```
mrinfo dwi.mif
*****
Image name:      "dwi.mif"
*****
Dimensions:      114 x 114 x 84 x 130
Voxel size:      1.79825 x 1.79825 x 1.8 x 3.1
Data strides:    [ -1 2 3 4 ]
Format:          MRtrix
Data type:        signed 16 bit integer (little endian)
Intensity scaling: offset = 0, multiplier = 1
Transform:        1 -3.457e-08 0.0002674 -102.9
                  3.077e-05 0.9934 -0.1149 -77.34
                  -0.0002656 0.1149 0.9934 -66.49
```

其中，Dimensions 记录图像的维度：前三维表示 XYZ 方向的体素数量，第四维

表示采集的 volume 数量。每个 volume 对应特定的梯度强度(bval)与方向(bvec)。因此，本例中 bval 包含 130 个 b 值，bvec 包含 130 个对应的梯度方向。

## 2. 去噪和去除 Gibbs 伪影: dwidenoise & mrdegibbs

通常，在进行其他 DWI 数据预处理步骤之前，需要先进行去噪（Denoising）和 Gibbs 伪影去除（Unringing）。

首先，使用 dwidenoise 命令进行去噪（Veraart et al. 2016）。该方法基于 Marchenko-Pastur 主成分分析(MP-PCA)，利用数据的局部冗余性识别并去除噪声成分：

```
dwidenoise dwi.mif dwi_den.mif -noise noise.mif
```

其中，dwi\_den.mif 为去噪后的图像，noise.mif 为估计的噪声图。

随后，使用 mrdegibbs 命令（Kellner et al. 2016）去除由 k 空间截断引起的 Gibbs 振铃伪影：

```
mrdegibbs dwi_den.mif dwi_den_unr.mif -axes 0,1
```

其中，dwi\_den\_unr.mif 为去除 Gibbs 伪影后的图像。-axes 参数用于指定数据采集平面：轴位为 0,1，冠状位为 0,2，矢状位为 1,2。本示例数据为轴位采集，因此设置为-axes 0,1。

## 3. 头动和 EPI 畸变校正: dwifslpreproc

MRtrix3 基于 FSL 进行头动校正和畸变校正，因此需安装 FSL 以确保该步骤顺利进行。本数据采集了反向编码图像，可采用 topup 工具进行畸变校正。

示例数据采集为 PA 方向编码。首先，使用 dwiextract 命令从 130 个 volume 中提取 b=0（b0 像）数据作为参考（b0\_PA.mif）：

```
dwiextract dwi_den_unr.mif b0_PA.mif -bzero
```

由于采集了多个 b0 像，可对其平均得到平均 b0 像（b0\_PA\_mean.mif），-axis 3 表示对第四维取平均：

```
mmrmath b0_PA.mif mean b0_PA_mean.mif -axis 3
```

示例数据集还采集了反向编码图像（sub-001\_acq-dwi\_dir-AP\_epi.nii）用于校正空间畸变，存放于 fmap 文件夹。首先将其转换为 MIF 格式（b0\_AP.mif）：

```
mrconvert sub-001_acq-dwi_dir-AP_epi.nii b0_AP.mif
```

随后提取反向编码的平均 b0 图像 (b0\_AP\_mean.mif):

```
mrmath b0_AP.mif mean b0_AP_mean.mif -axis 3
```

使用 mrcat 将两个方向的平均 b0 像拼接得到 b0\_pair.mif。主方向 (PA) 应在前，反向编码方向 (AP) 在后:

```
mrcat b0_PA_mean.mif b0_AP_mean.mif b0_pair.mif -axis 3
```

**注意:** 除拼接平均 b0 图像外，也可直接拼接 b0\_PA.mif 与 b0\_AP.mif，但需确保 b0\_pair.mif 的 volume 数量为偶数，前 N 个为 PA，后 N 个为 AP。本数据采集了 10 个 PA 编码和 5 个 AP 编码的 b0 图像，因此可从 b0\_PA.mif 中提取 5 个 volume 与 b0\_AP.mif 拼接。

然后，使用 dwifslpreproc 方法进行畸变校正，得到 dwi\_den\_unr\_preproc.mif:

```
dwifslpreproc dwi_den_unr.mif dwi_den_unr_preproc.mif \  
  -pe_dir PA \  
  -rpe_pair \  
  -se_epi b0_pair.mif \  
  -align_seepi \  
  -readout_time 0.0836197 \  
  -eddy_options "--slm=linear "
```

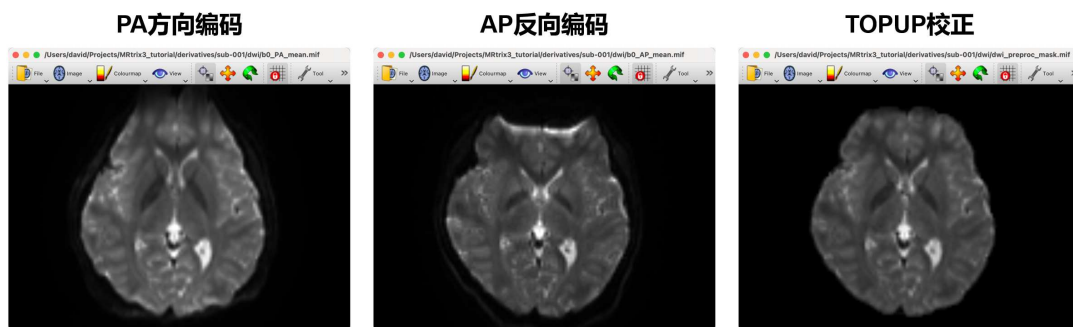
主要参数如下:

- -pe\_dir: 主相位编码方向，本例为 PA (Posterior→Anterior)
- -rpe\_pair: 表示存在反向编码图像，用于 topup 校正 EPI 畸变
- -se\_epi: 反向编码 b0 图像对 (b0\_pair.mif)，用于 topup 校正畸变
- -align\_seepi: 对反向编码 b0 图像对进行配准，改善 topup 畸变校正精度
- -readout\_time: 总读出时间 (total readout time, 单位秒)
- -eddy\_options "--slm=linear ": 使用线性模型进行 slice-to-volume 运动校正

**注意:** 由于 volume 内的层间校正运算量大，FSL 的 OpenMP 版本的 eddy 功能被简化，即使指定层间校正参数也不可用。仅 CUDA 版本的 eddy 支持完整功能。MRtrix3 会根据系统自动选择可用版本运行。

若运行出现 data is not shelled 错误，可添加参数:

```
-eddy_options "--data_is_shelled"
```



可见，校正前 PA 方向编码图像存在拉伸，而 AP 方向编码图像存在压缩；经过 topup 校正后，图像得到修正。

若未采集反向编码图像，则无法使用 topup 进行畸变校正，此时命令如下：

```
dwifslpreproc dwi_den_unr.mif dwi_den_unr_preproc.mif -pe_dir PA -rpe_none
```

#### 4. 偏差场校正（可选）：dwibiascorrect

使用 dwibiascorrect 进行偏差场校正(Bias Field Correction)为后续大脑掩膜(Brain Mask)的估计提供更可靠的数据基础。可采用 ants 方法以获得更精确的校正效果。该方法需先安装 ANTs，操作如下：

```
dwibiascorrect ants dwi_den_unr_preproc.mif dwi_den_unr_preproc_unbiased.mif  
-bias bias.mif
```

得到的 dwi\_den\_unr\_preproc\_unbiased.mif 为校正后的图像，bias.mif 为估计的偏差场图。

**注意：**该步骤不一定能改善 Mask 的估计效果，可与未执行该步骤（使用 dwi\_den\_unr\_preproc.mif 生成的 Mask）进行对比，以判断是否需要执行。

#### 5. 估计 Brain Mask：dwi2mask

使用 dwi2mask 命令从 DWI 数据自动生成 Brain Mask，操作如下：

```
dwi2mask dwi_den_unr_preproc_unbiased.mif dwi_mask.mif
```

由于本数据采集了空间分辨率更高的 T1w 图像，后续将基于 T1w 图像估计更精确的 Brain Mask。

### 3.3.4 T1w 结构像预处理

由于后续的 dMRI 数据分析涉及 T1w 结构像的使用，在此先对 T1w 结构像的预处理过程进行简要介绍。相关步骤主要使用 ANTs 工具完成，参数说明可参考 ANTs 官方文档，在此不做详细介绍。

#### 1. 偏差场校正：N4BiasFieldCorrection

首先，使用 N4BiasFieldCorrection 命令对 T1w 结构像（sub-001\_run-1\_T1w.nii）进行偏差场校正，以去除磁场不均匀性造成的信号强度偏差，得到校正后的图像 sub-001\_T1w\_n4.nii.gz：

```
N4BiasFieldCorrection \
  -d 3 \
  -i sub-001_run-1_T1w.nii \
  -o sub-001_T1w_n4.nii.gz \
  -s 4 \
  -b [200] \
  -c [50x50x50x50,0.0000001]
```

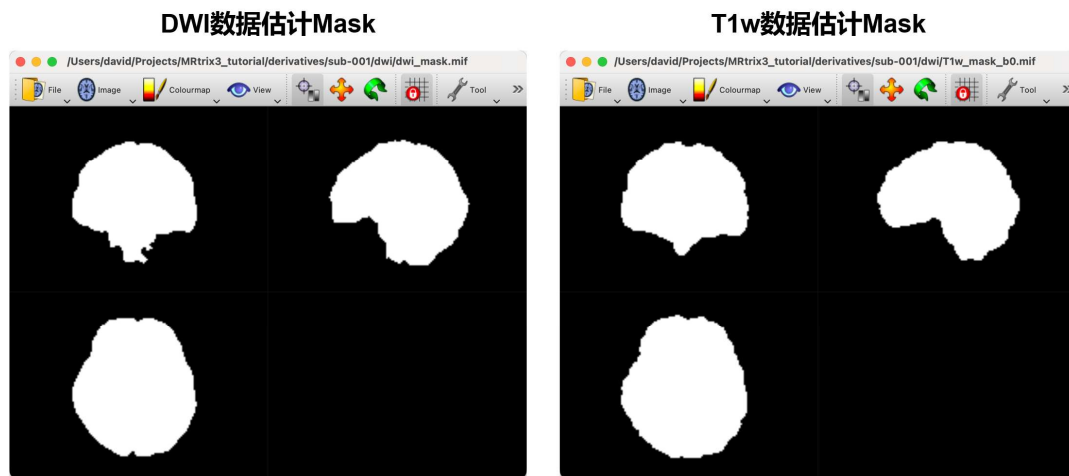
#### 2. 颅骨去除：antsBrainExtraction

然后，使用 antsBrainExtraction.sh 命令，从 T1w 结构像中提取脑组织，去除颅骨、头皮等非脑组织部分。此处采用 OASIS 模板作为目标模板（下载链接：[https://figshare.com/articles/dataset/ANTs\\_ANTsR\\_Brain\\_Templates/915436?file=3133832](https://figshare.com/articles/dataset/ANTs_ANTsR_Brain_Templates/915436?file=3133832)）。命令如下：

```
antsBrainExtraction.sh \
  -d 3 \
  -a sub-001_T1w_n4.nii.gz \
  -e T_template0.nii.gz \
  -m T_template0_BrainCerebellumProbabilityMask.nii.gz \
  -f T_template0_BrainCerebellumRegistrationMask.nii.gz \
  -o sub-001_T1w_
```

输出文件包括：

- sub-001\_T1w\_BrainExtractionBrain.nii.gz: 去颅骨后的脑组织图像
- sub-001\_T1w\_BrainExtractionMask.nii.gz: 对应的脑组织掩膜 (Brain Mask)



### 3. 配准到 b0 空间: flirt

由于 T1w 结构像和 DWI 的 b0 像所处空间不同, 需将两者进行配准。

首先, 从预处理后的 DWI 图像提取 b0 像 (b0\_PA\_preproc.mif):

```
dwextract dwi_den_unr_preproc_unbiased.mif b0_PA_preproc.mif -bzero
```

然后, 对预处理后的 b0 像进行平均 (b0\_PA\_preproc\_mean.mif):

```
mrmath b0_PA_preproc.mif mean b0_PA_preproc_mean.mif -axis 3
```

将平均 B0 转换为 NIFTI 格式以供 FSL 使用:

```
mrconvert b0_PA_preproc_mean.mif b0_PA_preproc_mean.nii.gz
```

使用 FSL 的 flirt 命令, 将偏差场校正并去颅骨后的 T1w 结构像配准至预处理后的平均 b0 像, 得到 b0 空间下的 T1w 像 (T1w\_Brain\_b0.nii.gz):

```
flirt -in sub-001_T1w_BrainExtractionBrain.nii.gz \
      -ref b0_PA_preproc_mean.nii.gz \
      -out T1w_Brain_b0.nii.gz \
      -omat T1w_to_b0.mat \
      -dof 6
```

对于同一被试不同模态影像间的配准, 采用线性的仿射变换即可 (6 个自由度)。

其中, T1w\_to\_b0.mat 记录了从个体 T1 空间到个体 b0 空间的仿射变换矩阵。

**注意:** 使用去颅骨后的 T1w 结构像可以更准确地与 b0 像进行配准, 从而得到变

现参数（即仿射变换矩阵）。

利用该仿射变换矩阵，可将带颅骨的 T1w 像配准到 b0 空间，得到 T1w\_b0.nii.gz：

```
flirt -in sub-001_T1w_n4.nii.gz \
      -ref b0_PA_preproc_mean.nii.gz \
      -out T1w_b0.nii.gz \
      -init T1w_to_b0.mat \
      -applyxfm
```

也可将 T1w 像的 Brain Mask 配准到 b0 空间，得到 T1w\_mask\_b0.nii.gz：

```
flirt -in sub-001_T1w_BrainExtractionMask.nii.gz \
      -ref b0_PA_preproc_mean.nii.gz \
      -out T1w_mask_b0.nii.gz \
      -init T1w_to_b0.mat \
      -applyxfm \
      -interp nearestneighbour
```

**注意：**对于 Mask/Atlas 等标签图像，插值算法应选择最近邻插值。

转换为 MIF 格式以供 MRtrix3 使用：

```
mrconvert T1w_mask_b0.nii.gz T1w_mask_b0.mif
```

将 Mask 应用于预处理后的 DWI 数，得到 dwi\_preproc\_masked.mif：

```
mrcalc dwi_den_unr_preproc_unbiased.mif T1w_mask_b0.mif -mult
dwi_preproc_masked.mif
```

在后续的 DWI 数据分析中，若无 T1w 结构像，则使用 dwi\_mask.mif；若有 T1w 结构像，则使用 T1w\_mask\_b0.mif，可提供更准确的 Brain Mask。

#### 4. 空间标准化：antsRegistrationSyNQuick.sh

最后，使用 antsRegistrationSyNQuick.sh 将去颅骨后的个体 T1w 图像配准到标准空间下的去颅骨模板——ICBM 152 Nonlinear Asymmetrical template version 2009c（即 MNI 2009c 模板）：

```
antsRegistrationSyNQuick.sh \
      -d 3 \
```



```
-f tpl-MNI152NLin2009cAsym_res-01_desc-brain_T1w.nii.gz \
-m sub-001_T1w_BrainExtractionBrain.nii.gz \
-o sub-001_T1w_to_MNI_ \
-t s
```

输出文件包括：

- sub-001\_T1w\_to\_MNI\_Warped.nii.gz: 个体图像在 MNI 空间下的结果
- sub-001\_T1w\_to\_MNI\_0GenericAffine.mat: 仿射变换矩阵
- sub-001\_T1w\_to\_MNI\_1Warp.nii.gz: 非线性形变场
- sub-001\_T1w\_to\_MNI\_1InverseWarp.nii.gz: 逆变形场（用于从 MNI 空间回到个体空间）

### 3.3.5 白质微结构计算

在完成 DWI 数据的预处理后，可以进行扩散张量成像建模，并从扩散张量计算各种标量和矢量指标量化白质微结构特性。

#### 1. 扩散张量估计: **dwi2tensor**

使用 **dwi2tensor** 命令进行张量建模，得到 **tensor.mif**：

```
dwi2tensor dwi_preproc_masked.mif tensor.mif -mask T1w_mask_b0.mif
```

然后，使用 **tensor2metric** 计算白质微结构特征 FA、MD、AD 和 RD：

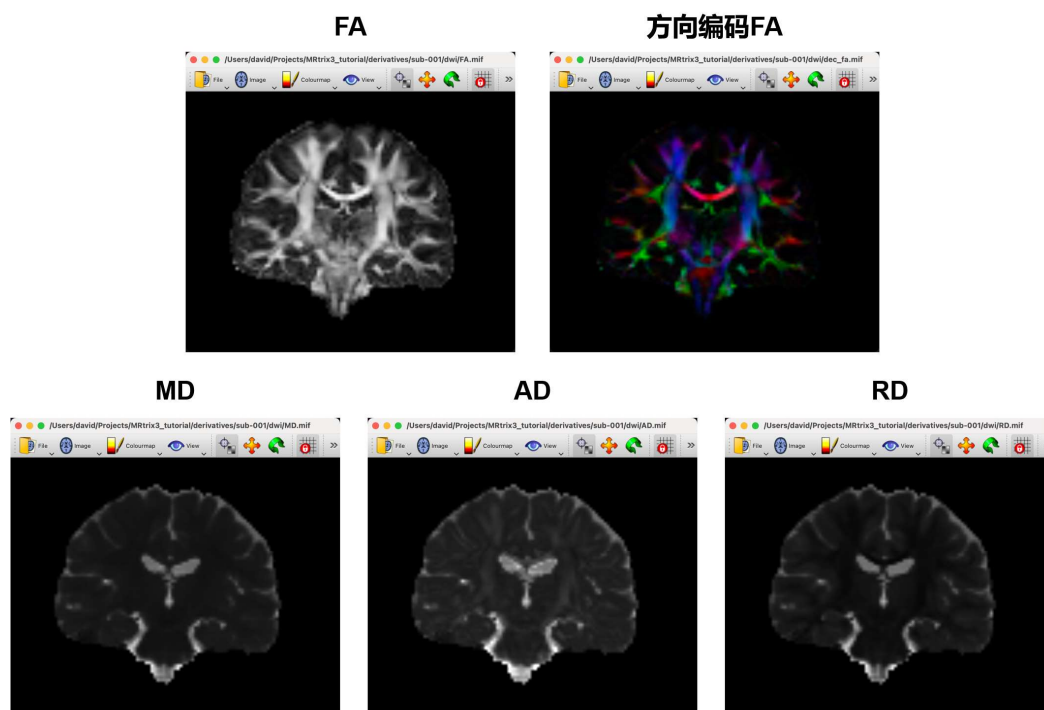
```
tensor2metric tensor.mif \
-fa FA.mif \
-adc MD.mif \
-ad AD.mif \
-rd RD.mif \
-mask T1w_mask_b0.mif \
-vector vector.mif
```

其中，**-vector** 参数生成 DTI 中第一主特征向量的图像。将 **vector.mif** 与 FA 值结合可生成方向编码的彩色图 **dec\_fa.mif**（红色-左右，绿色-前后，蓝色-上下）：

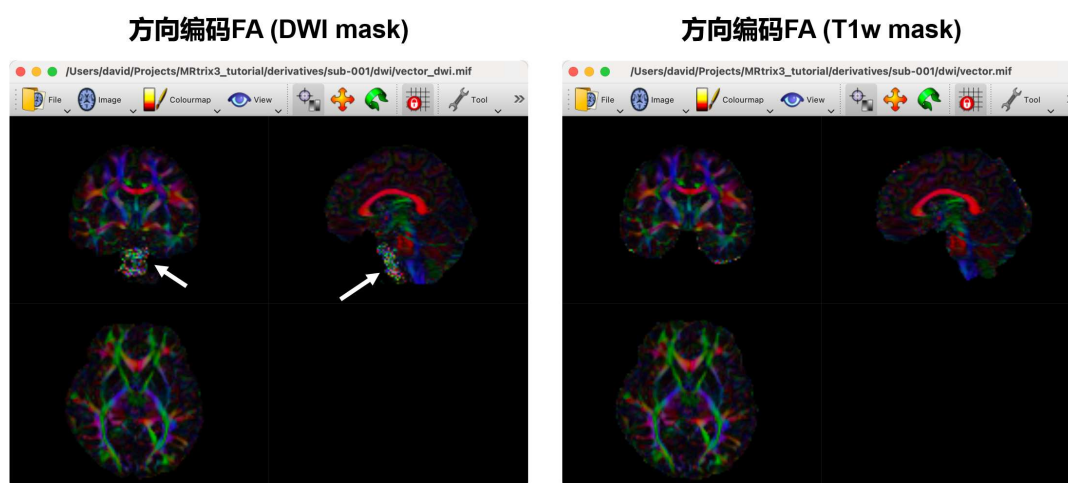
```
mrcalc vector.mif -abs FA.mif -mult dec_fa.mif
```

可使用 mrview 查看方向编码的 FA 图像 dec\_fa.mif:

```
mrview dec_fa.mif
```



此外，对比分别在 DWI 与 T1w 数据的 Brain Mask 下生成的方向编码 FA 图像可见，基于 DWI 数据估计的 Brain Mask 可能包含更多噪声信号。因此，建议使用空间分辨率更高的 T1w 结构像估计 Brain Mask。



### 3.3.6 纤维方向分布函数估计

在进行白质纤维追踪前，首先需要估计纤维方向分布函数（Fiber Orientation Distribution, FOD）。大致包括估计响应函数、估计 FOD 和强度标准化三个步骤。

为了更好地估计交叉纤维，可使用约束球形反卷积（Constrained Spherical Deconvolution, CSD）方法（Tournier et al., 2004, Tournier et al., 2008），取代普通的张量模型。

### 1. 响应函数估计：dwi2response

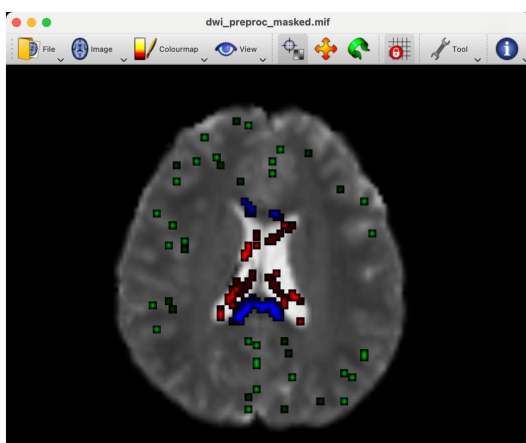
使用 `dwi2response` 命令估计白质（`wm_response.txt`）、灰质（`gm_response.txt`）和脑脊液（`csf_response.txt`）三种组织类型的响应函数，作为 CSD 建模的先验信息：

```
dwi2response dhollander dwi_preproc_masked.mif \
  wm_response.txt \
  gm_response.txt \
  csf_response.txt \
  -voxels RF_voxels.mif
```

此处选择 `dhollander` 算法，通过无监督学习自动识别单纤维白质体素、灰质体素和 CSF 体素，分别估计其扩散响应特征。该方法基于扩散信号的球谐系数特征进行组织分类。此外，MRtrix3 还提供其他算法，如 `fa`、`msmt_5tt`、`tournier` 等，具体说明可参见官方文档。

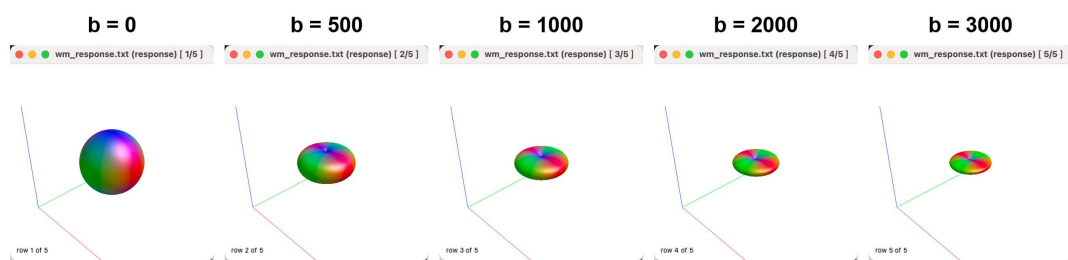
其中，`RF_voxels.mif` 标记了在估计 WM、GM、CSF 响应函数时实际使用的体素，可用于质量控制，检查算法是否正确选择了体素（红色体素应位于 CSF，蓝色体素应位于 WM，绿色体素应位于 GM）：

```
mrview dwi_preproc_masked.mif -overlay.load RF_voxels.mif
```



此外，可使用 `shview` 查看各组织的响应函数。以白质响应函数为例：

```
shview wm_response.txt
```



本数据包含 5 个  $b$  值 ( $b=0, 500, 1000, 2000, 3000$ )，对应 5 种响应函数。可观察到，随着  $b$  值增大，响应函数逐渐变得扁平。

## 2. 估计 FOD: dwi2fod

基于前一步获得的各组织响应函数，使用 `dwi2fod` 命令估计白质 (`wmfod.mif`)、灰质 (`gm.mif`) 和脑脊液 (`csf.mif`) 的 FOD 函数：

```
dwi2fod msmt_csd dwi_preproc.mif \
    -mask T1w_mask_b0.mif \
    wm_response.txt wmfod.mif \
    gm_response.txt gm.mif \
    csf_response.txt csf.mif
```

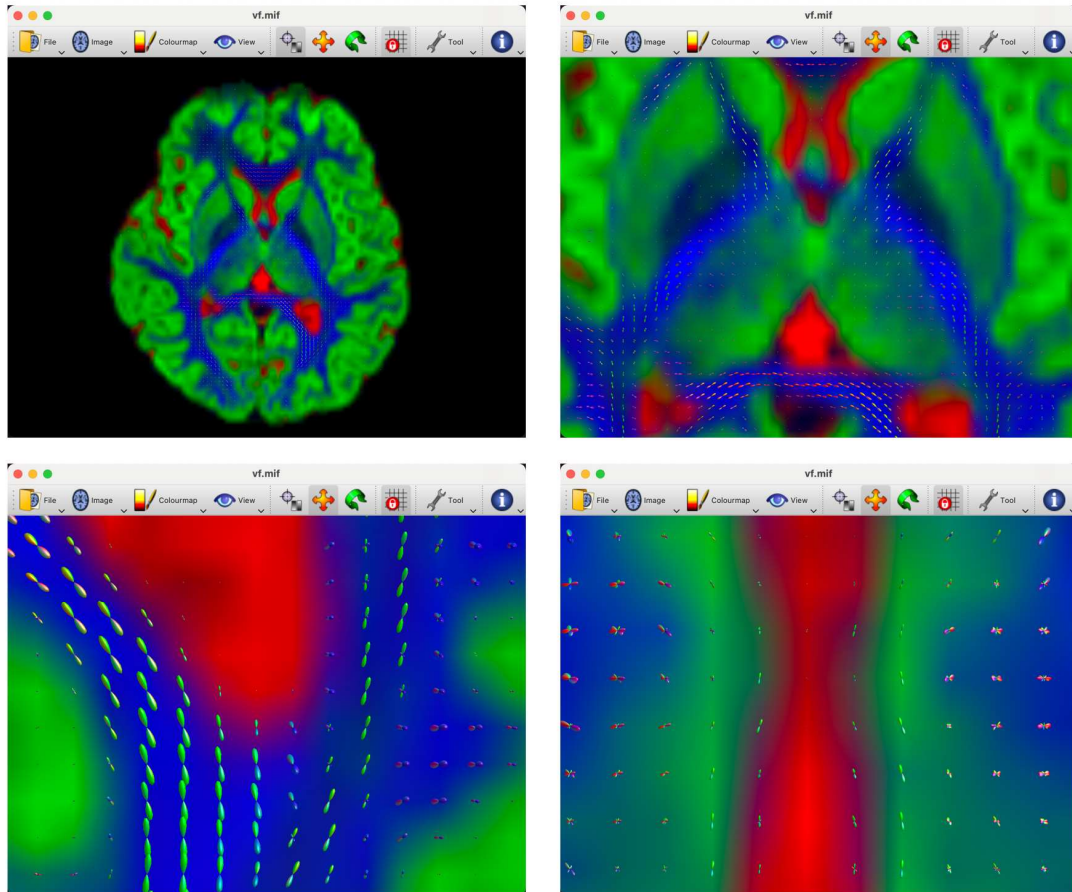
此处采用 `msmt_csd` 算法 (Multi-Shell Multi-Tissue Constrained Spherical Deconvolution, MSMT-CSD)，即多壳多组织约束球形反卷积 (Dhollander et al., 2019, Dhollander, Raffelt, and Connelly, 2016)。该方法利用不同组织类型对不同  $b$  值的特征响应，联合估计多组织的方向分布函数。

为检查 `wmfod` 的估计质量，尤其是在存在交叉纤维区域时的表现，可将其叠加显示在解剖底图上。底图可使用组织体积分数图 (`volume fraction, vf.mif`)，表示各组织在体素中的相对体积分布：

```
mrconvert -coord 3 0 wmfod.mif - | mrcat csf.mif gm.mif - vf.mif
```

随后，将 `wmfod.mif` 叠加于 `vf.mif` 进行可视化：

```
mrview vf.mif -odf.load_sh wmfod.mif
```



可观察到，在交叉纤维区域，FOD 呈多峰分布；而在单纤维区域，FOD 主要呈单一方向性，结果合理反映了白质纤维的空间结构特征。

### 3. 强度标准化：mtnormalise

使用 mtnormalise (Raffelt et al., 2017) 对个体 FOD 的全局强度进行标准化，得到强度标准化后的白质 (wmfod\_norm.mif)、灰质 (gm\_norm.mif) 和脑脊液 (csf\_norm.mif) 的 FOD 函数，用于后续跨被试的组水平统计分析：

```
mtnormalise wmfod.mif wmfod_norm.mif \
  -mask T1w_mask_b0.mif \
  gm.mif gm_norm.mif \
  csf.mif csf_norm.mif
```

#### 3.3.7 纤维追踪

在得到白质的 FOD 后，即可进行纤维追踪得到全脑纤维束图。主要步骤包括解

剖约束纤维追踪、全脑纤维追踪和球面反卷积引导的纤维追踪滤波。

## 1. 解剖约束纤维追踪: 5ttgen & 5tt2gmwmi

解剖约束追踪（Anatomically Constrained Tractography, ACT）框架在进行纤维追踪时，将结构 MRI 提供的解剖信息纳入约束，使追踪结果更符合生物学实际。例如：纤维应起止于灰质或白质交界处；纤维不应穿越脑脊液区域；白质束应位于白质组织中。

使用 5ttgen 将 T1w 结构像分割为五种组织类型（T1w\_5tt.mif），包括皮层灰质、皮层下灰质、白质、脑脊液和病理组织：

```
5ttgen fsl T1w_b0.nii.gz T1w_5tt.mif -sgm_amyg_hipp -nocrop
```

5ttgen 支持 fsl、freesurfer、gif 和 hsvs 四种分割模式，此处选择 fsl 方法。参数 -sgm\_amyg\_hipp 用于将杏仁核和海马划为皮层下灰质，-nocrop 表示不裁剪输出图像。生成的 5TT 图像包括五个组织通道，其中病理组织部分通常为空白。

**注意：**此处应使用带颅骨的 T1w 结构像以确保组织的正确分割。

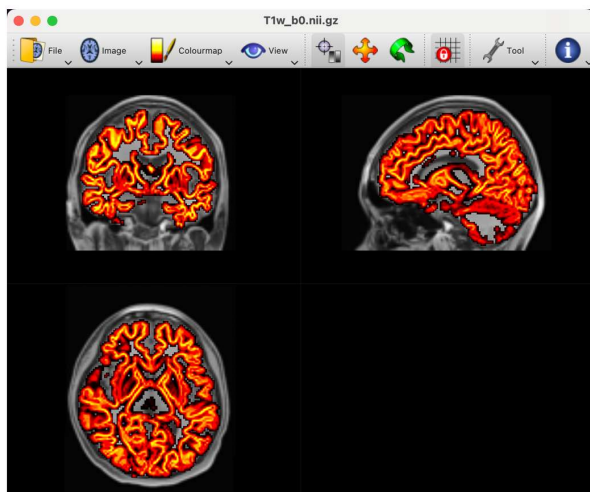


最后，使用 5tt2gmwmi 从 5TT 图像中提取灰白质交界面（gmwmSeed.mif），定义纤维追踪的种子区域，以提高追踪的解剖学准确性：

```
5tt2gmwmi T1w_5tt.mif gmwmSeed.mif
```

可将灰白质分界面叠加于 T1w 结构像进行检查：

```
mrview T1w_b0.nii.gz --overlay.load gmwmSeed.mif
```



## 2. 全脑纤维追踪: **tckgen**

在获得白质 FOD (wmfod\_norm.mif) 并提取灰白质交界面 (gmwmSeed.mif) 后, 即可使用 **tckgen** 命令进行全脑纤维追踪, 生成全脑纤维束图 (tracks\_10M.tck):

```
tckgen wmfod_norm.mif tracks_10M.tck \  
-act T1w_5tt.mif \  
-backtrack \  
-seed_gmwmi gmwmSeed.mif \  
-minlength 30 \  
-maxlength 250 \  
-cutoff 0.06 \  
-select 10M
```

**tckgen** 支持多种追踪算法, 默认使用 **iFOD2** 概率性算法, 可在复杂交叉纤维区域更稳定地生成合理的纤维路径。

主要参数如下:

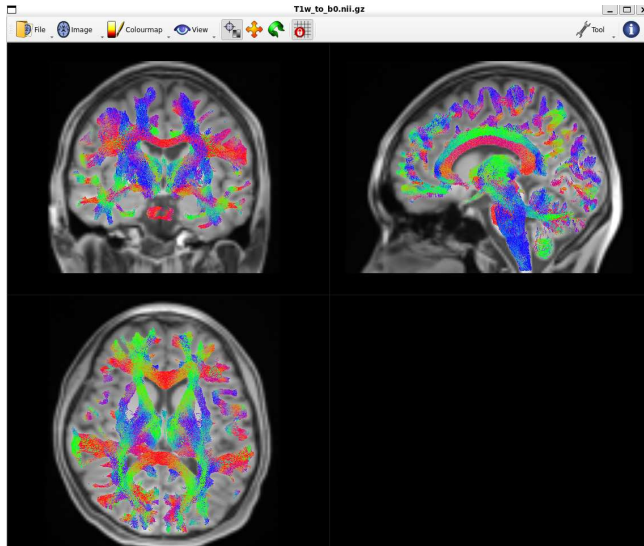
- **-act**: 解剖约束追踪, 使用 5TT 图像 (T1w\_5tt.mif)
- **-backtrack**: 启用回溯机制, 遇到无效区域时后退
- **-seed\_gmwmi**: 从灰白质交界面播种
- **-minlength**: 最小纤维长度 (30 mm)
- **-maxlength**: 最大纤维长度 (250 mm)
- **-cutoff**: FOD 振幅阈值 (0.6), 低于此值停止追踪



- **-select:** 目标纤维流线数量（通常需要大于 1000 万）

由于生成 10M 条 streamline 文件体积较大，直接加载不便。此处使用较小的子集（如 20 万条 streamline）进行可视化示例：

```
mrview T1w_b0.nii.gz -tractography.load tracks_200k.tck
```



### 3. 修正纤维权重: tcksift2

由于纤维追踪生成的 streamlines 数量不能直接反映真实的纤维密度，tcksift2（SIFT2 = Spherical-deconvolution Informed Filtering of Tractograms 2）通过为每根 streamline 分配合理权重，使得纤维束密度与 FOD 信号一致，显著提高了定量分析的准确性。相比简单地计数 streamlines，使用 SIFT2 权重能更真实地反映白质连接的强度。

```
tcksift2 -act T1w_5tt.mif tracks_10M.tck wmfod_norm.mif sift_10M.txt
```

该命令基于解剖约束和强度标准化后的白质 FOD，对已生成的 1000 万条纤维进行筛选与加权，输出结果 sift\_10M.txt 记录了每条 streamline 的权重系数  $w$ ：

- $w \approx 1$ ：该纤维的密度与 FOD 信号预测一致；
- $w > 1$ ：该区域的纤维数偏少，需要“加权”增强；
- $w < 1$ ：该区域的纤维数偏多，需要“降权”；
- $w \approx 0$ ：该纤维可能为伪迹，对整体连接贡献极小。

通过 SIFT2 校正后，得到的加权纤维结果可更准确地用于后续的结构连接矩阵构建及跨被试比较分析。



### 3.3.8 构建结构连接组

结构连接组 (structural connectome), 又称结构网络 (structural network), 由节点 (nodes) 和连边 (edges) 组成。其中, 节点通常定义为预先分区的脑区, 连边的权重反映脑区间白质连接的强度。

#### 1. 脑图谱配准到个体 b0 空间

在本示例中, 选用 Schaefer 图谱 (Schaefer2018\_400Parcels\_7Networks) 中 400 个皮层分区作为网络节点。由于该图谱位于 MNI 标准空间, 需要依次将其配准到个体 T1 空间, 再转换到个体 dMRI 的 b0 空间。

首先, 使用 ANTs 的 `antsApplyTransforms` 命令, 将 MNI 空间下的 Schaefer 图谱配准到个体 T1 空间。该过程使用前面步骤中生成的从 T1 空间到 MNI 空间的逆形变场和仿射矩阵:

```
antsApplyTransforms \  
  -d 3 \  
  -i Schaefer2018_400Parcels_7Networks_order_FSLMNI152_1mm.nii.gz \  
  -r sub-001_T1w_BrainExtractionBrain.nii.gz \  
  -o schaefer400.nii.gz \  
  -n NearestNeighbor \  
  -t sub-001_T1w_to_MNI_1InverseWarp.nii.gz \  
  -t [sub-001_T1w_to_MNI_0GenericAffine.mat,1]
```

**注意:** 对脑图谱进行变形时, 应使用最近邻插值 (`-n NearestNeighbor`) 以确保其数值反应原本的标签信息。

接下来, 使用 FSL 的 `flirt` 命令, 将 T1 个体空间下的图谱进一步配准至 dMRI 的 b0 空间。这里使用前面配准步骤中获得的 T1w 到 b0 的仿射矩阵 (`T1w_to_b0.mat`):

```
flirt \  
  -in schaefer400.nii.gz \  
  -ref b0_PA_preproc_mean.nii.gz \  
  -out schaefer400_b0.nii.gz
```

```
-init T1w_to_b0.mat \  
-applyxfm \  
-interp nearestneighbour
```

同样地，保持最近邻插值方式以保证标签完整性（-interp nearestneighbour）。最后，将结果转换为 MRtrix 格式，以便后续分析：

```
mrconvert schaefer400_b0.nii.gz schaefer400_b0.mif
```

## 2. 构建结构连接矩阵：tck2connectome

基于前面获得的全脑纤维追踪结果（tracks\_10M.tck）、streamline 权重系数（sift\_10M.txt）以及 b0 空间下的脑图谱，可使用 tck2connectome 命令构建个体的结构连接矩阵（sc\_schaefer400\_10M.csv）：

```
tck2connectome \  
-symmetric \  
-zero_diagonal \  
-scale_invnodevol \  
-assignment_radial_search 2 \  
-tck_weights_in sift_10M.txt \  
tracks_10M.tck \  
schaefer400_b0.mif \  
sc_schaefer400_10M.csv \  
-out_assignment assignments_schaefer400_10M.csv
```

主要参数如下：

- -symmetric：生成对称矩阵（无向图）
- -zero\_diagonal：对角线设为 0（去除自连接）
- -scale\_invnodevol：按节点体积的倒数归一化，减少脑区体积差异的影响
- -assignment\_radial\_search 2：设置径向搜索半径为 2 mm，用于确定流线端点所属脑区
- -tck\_weights\_in：输入 SIFT2 计算得到的权重文件，使连接强度更符合 FOD 信号分布

- `-out_assignment`: 输出每条流线对应的脑区配对信息，便于后续追踪分析
- 输出文件包括：
- `sc_schaefer400_10M.csv`:  $400 \times 400$  结构连接矩阵，矩阵元素  $(i, j)$  表示脑区  $i$  与  $j$  之间的结构连接强度
  - `assignments_schaefer400_10M.csv`: 记录每条流线两端连接的脑区编号及其权重信息，可用于进一步分析连接分布与特征

至此，基于 MRtrix3 的 DWI 数据预处理与分析的主要流程已介绍完毕。更多高级分析方法（如 Fixel-Based Analysis）及各命令的详细参数说明，可参考官方文档：<https://mrtrix.readthedocs.io/en/latest/index.html>。若在实际操作中遇到问题，可前往 MRtrix3 官方社区查阅或提问：<https://community.mrtrix.org/c/wiki/12>。