DTI data processing

目录

[PART 1 整理数据 2](#_Toc141880180)

[PART 2 DTI数据预处理 6](#_Toc141880181)

[PART 3 TBSS 数据处理 (FA, 组分析) 7](#_Toc141880182)

[PART 4 TBSS 数据处理（non-FA images：MD, 组分析） 14](#_Toc141880183)

[PART 5 相关分析 (FA) 18](#_Toc141880184)

[PART 6 相关分析 (MD) 22](#_Toc141880185)

[PART 7 提取感兴趣纤维束（ROIs） 25](#_Toc141880186)

# PART 1 整理数据

1. 原始数据文件夹命名
2. 把全部被试的原始数据存放在一个文件夹 （e.g., DTI\_YI/RawData/DTI）
3. 再对每个被试（子文件夹）进行统一命名

* 比如，subj01， subj02, subj03 …
* 或者，Caiziyi, Cheyibo, Denghao …

\* 统一大小写，避免出现被试顺序混乱!

1. 转换 dcm数据格式

创建/修改脚本文件：

cd /DTI\_YI/Scripts/DTI

gedit dicom2nii\_DTI.sh &

#!/bin/bash

datapath=/home/ld/DTI\_YI

subjid=`ls $datapath/RawData/DTI | sed -n '1,8p'`

# or as below: for a specific subject

# subjid = ‘Caiziyi’

for subj in $subjid

do

echo $subj

mkdir -p $datapath/Converted/BOLD/$subj

dcm2nii -t y -o $datapath/Converted/BOLD/$subj $datapath/RawData/DTI /$subj

dcm2niix -o $datapath/Converted/BOLD/$subj -z y $datapath/RawData/DTI/$subj

done

\*根据当前数据修改路径

1. 整理被试信息到 excel 表格，并复制转换后的数据到新文件夹

cd Converted/BOLD/subj

1. 定位目标文件：找到：

a.以‘被试名’开头的（而不是以‘日期’）、

b.同时拥有 ‘.bval’ ‘.bvec’ ‘.json’为后缀的（只有. json 的不行）、

c.DTI 文件(不是t1)。

比如，下图红色方框标记的文件

A screenshot of a computer

Description automatically generated with medium confidence

1. 定位txt文件：根据第 1）步所找文件的末尾数字（比如上图为 10 和 12），找到以‘日期’开头的文件。比如，下方绿色方框标记的文件

A screenshot of a computer code

Description automatically generated with low confidence

3）查看 txt文件内容:比如第2步中末尾数字为 10和 12的 txt文件:

cat 20221113\_143030ep2ddiffmddw20p2caiziyis010a000.txt

cat 20221113\_143030ep2ddiffmddw20p2caiziyis012a1001.txt

A picture containing text, screenshot, font

Description automatically generated

找到DimXYZT最后一个维度为 72（而不是 1） 的文件，上方例子的 10。

4） 复制txt.里的被试信息到 excel 表格

表格格式如图所示：其中，Scan date为 txt文件名的前缀

A picture containing text, font, screenshot, number

Description automatically generated

5） 将数据复制到新文件夹

第 3）步中，确认以 10 结尾的文件为目标 DTI 文件。

将a.以‘被试名’开头的、b.以 10 结尾的 全部文件复制到新文件夹里。

比如，上方例子中，即为第1个红色方框标记的全部文件.

先创建目标文件夹(DTI\_YI/DTI/TD/Caiziyi)：

cd DTI\_YI

mkdir DTI

cd DTI

mkdir TD # (比如，分为 TD 和 ASD 两组数据时)

cd TD

mkdir Caiziyi

再复制文件：

cd DTI\_YI/Converted/BOLD/Caiziyi

cp Caiziyi\_ep2d\_diff\_mddw\_20\_p2\_20221113143031\_10.\* DTI\_YI/DTI/TD/Caiziyi

# PART 2 DTI数据预处理

参考已有脚本文件：~/DTI\_YI/Scripts/ DTI\_preprocess.sh

#!/bin/sh

#preprocessing

path=/home/ld/DTI\_YI/DTI/ASD

subjlist=`ls $path` #| sed -n '1,1p'`

#subjlist=`ls Xujiayi Hehaoyang`

for subj in $subjlist

do

# rename files

cd $path/$subj

if [ ! -f dti\_FA.nii.gz ]

then

file=`ls . | grep 'nii.gz'`

bvecs=`ls . | grep 'bvec'`

bvals=`ls . | grep 'val'`

mv $file DWI.nii.gz

mv $bvecs DWI.bvec

mv $bvals DWI.bval

# Step 1: eddy correction

eddy\_correct DWI.nii.gz DWI\_eddy.nii.gz 0

# Step 2: correct the direction of b matrix

#bash fdt\_rotate\_bvecs DWI.bvec DWI\_rotate.bvec DWI\_eddy.ecclog

# Step 3: mask file

fslroi DWI\_eddy.nii.gz b0.nii.gz 0 1

bet2 b0.nii.gz b0\_brain -f 0.2 -m

# Step 4: DTI

dtifit -k DWI\_eddy.nii.gz -m b0\_brain\_mask.nii.gz -o dti -r DWI.bvec -b DWI.bval

fi

done

每个被试文件夹里生成以下文件：

A screen shot of a computer code

Description automatically generated with medium confidence

# PART 3 TBSS 数据处理 (FA, 组分析)

1. 复制 FA或 MD 文件到单独的文件夹，并对文件重命名
2. 先创建 txt文件夹，存放ASD 和 TD 各组内的被试名。比如，ASD组

A screenshot of a computer

Description automatically generated with medium confidence

2）参考已有脚本文件，进行复制与重命名：~/DTI\_YI/Scripts/cp\_FA.sh

#!/bin/sh

# copy FA files

path=/home/ld/DTI\_YI/DTI

# ASD

subj\_ASD=`cat /home/ld/DTI\_YI/Scripts/files/subj\_ASD\_DTI.txt`

for subj in $subj\_ASD

do

cp $path/ASD/$subj/dti\_FA.nii.gz $path/TBSS/${subj}\_FA.nii.gz

# TD

subj\_TD=`cat /home/ld/DTI\_YI/Scripts/files/subj\_TD\_DTI.txt`

for subj in $subj\_TD

do

cp $path/TD/$subj/dti\_FA.nii.gz $path/TBSS/${subj}\_FA.nii.gz

done

1. TBSS处理
2. 参考已有脚本文件：~/DTI\_YI/Scripts/tbss.sh

可参照脚本分步运行，以便确认每一步生成哪些文件，以及可能在哪一步出问题。

脚本细节可以参考 TBSS/UserGuide：<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/TBSS/UserGuide>

*Step1 & 2 : tbss\_1\_preproc \* 和 tbss\_2\_reg -n*

因为第二步需要很长时间，所以可以将这两步单独创建脚本文件，在后台运行。

* cd ~/DTI\_YI/Scripts
* gedit tbss2\_reg.sh &

#!/bin/sh

path=/home/ld/DTI\_YI/DTI/TBSS/FA

cd $path

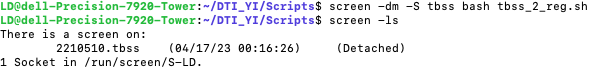
# pre-processing: remove brain-edge artifacts and zero the end slicess (inside the TBSS folder)

#tbss\_1\_preproc \*

# Registering all the FA data

tbss\_2\_reg -n

* bash
* screen -dm -S tbss bash tbss\_2\_reg.sh
* screen -ls



tbss\_1\_preproc \* 后生成 origdata & FA两个文件夹

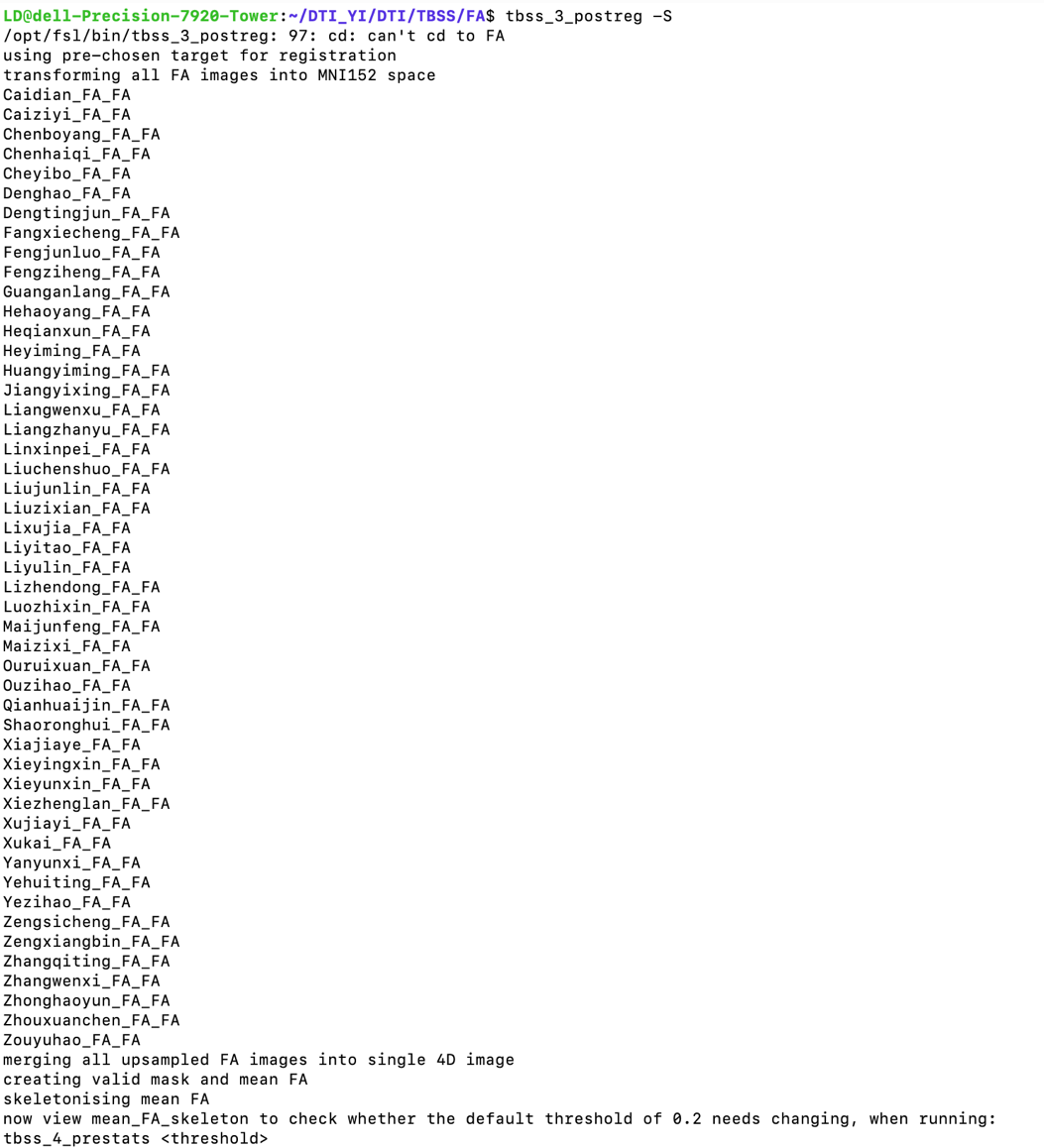
Origdata: 储存原始的subj\_FA.nii.gz文件

FA: slicesdir储存所有png图像和index.html

tbss\_2\_reg -n 后，每个被试会生成\*warp.nii.gz 文件，用于第三步。

*Step 3:* *tbss\_3\_postreg -S*

成功运行后的结果：

****

这一步可能出现的问题是，部分被试没有互相配对成功，会弹出类似下图的提示。利用提示找到 best target，进行所有被试和 best target 的匹配（即修改tbss\_2\_reg.sh文件后，重新运行）。结束后需要确认所有被试都和 best target生成匹配的subj\_FA\_FA\_to\_besttarget\_FA\_FAwarp.nii.gz文件。最后再次重新运行 step3 : *tbss\_3\_postreg -S*

* gedit tbss2\_reg.sh &

#!/bin/sh

path=/home/ld/DTI\_YI/DTI/TBSS/FA

cd $path

# pre-processing: remove brain-edge artifacts and zero the end slicess (inside the TBSS folder)

#tbss\_1\_preproc \*

# Registering all the FA data

tbss\_2\_reg -t S34\_dti\_FA\_FA.nii.gz

* bash
* screen -dm -S tbss bash tbss\_2\_reg.sh
* screen -ls

**Text

Description automatically generatedText

Description automatically generated**

*Step 4: tbss\_**4\_prestats 0.2*

*>* cd stats

*>* tbss\_*4\_prestats 0.2*

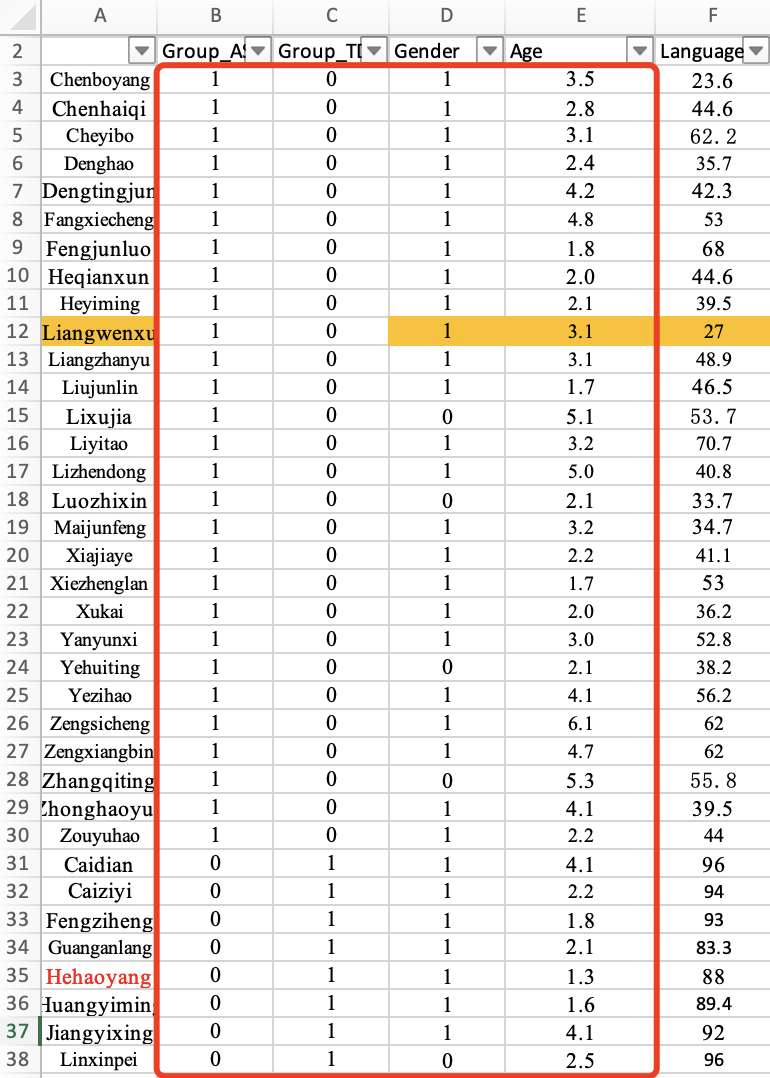
**A picture containing text, screenshot, font, algebra

Description automatically generated**

*Step 5: randomize （for group comparison）*

首先在 TBSS/stats 路径下创建 contrast 和 matrix 文件（randomise, FDR correction, tbss fill时用到），可参考GLM/CreatingDesignMatricesByHand：<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/GLM/CreatingDesignMatricesByHand>

创建 excel 表格存放所需数据，如



用红色方框里的内容创建文件design.txt

contrast.txt



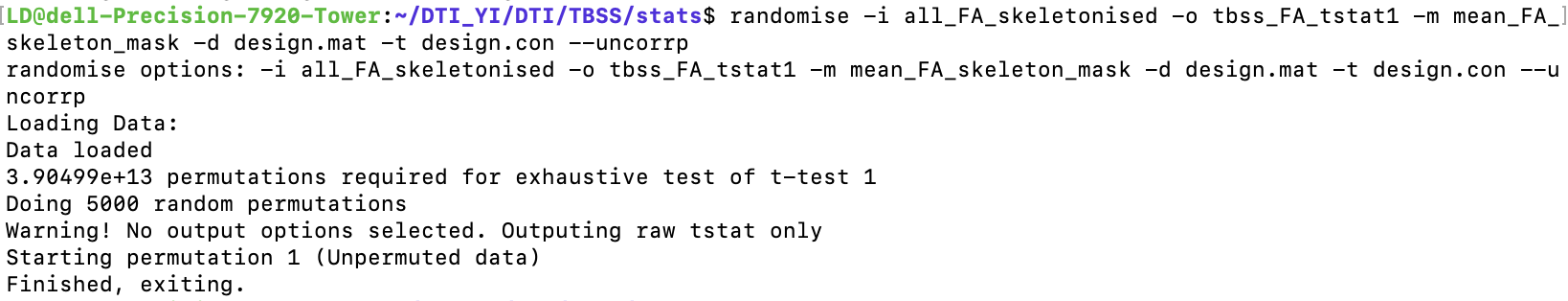
将design.txt 和 contrast.txt 分别转换为：design.mat和 design.con：

Text2Vest design.txt design.mat

Text2Vest contrasts.txt design.con

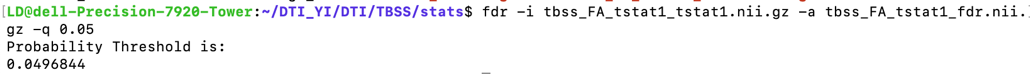
# randomize (for group comparison)

randomise -i all\_FA\_skeletonised -o tbss\_FA\_tstat1\_tstat1 -m mean\_FA\_skeleton\_mask -d design.mat -t design.con --uncorrp



# FDR correction (for group comparison)

fdr -i tbss\_FA\_tstat1.nii.gz -a tbss\_FA\_tstat1\_fdr.nii.gz -q 0.05



# tbss fill (这一步如果无法打开 fsleyes, 可以打开 matlab -nodeskop 后用 DPABI\_VIEW 里查看)

tbss\_fill tbss\_tfce\_corrp\_tstat1 0.95 mean\_FA tbss\_fill\_FA

fslview mean\_FA -b 0,0.6 mean\_FA\_skeleton -l Green -b 0.2,0.7 tbss\_fill\_FA -l Red-Yellow

# PART 4 TBSS 数据处理（non-FA images：MD, 组分析）

步骤与 FA 基本一致以 MD (mean diffusivity) 为例：

1. 创建新文件夹

>cd TBSS

>mkdir MD

1. 复制 MD 图像到此文件夹，并重命名：名字要和 FA 文件保持一致（即使此处时 MD 图像，命名里仍为 FA）

参考脚本：~/DTI\_YI/Scripts/cp\_FA.sh

#!/bin/sh

# copy FA/MD files

path=/home/ld/DTI\_YI/DTI

# ASD

subj\_ASD=`cat /home/ld/DTI\_YI/Scripts/files/subj\_ASD\_DTI.txt`

for subj in $subj\_ASD

do

cp $path/ASD/$subj/dti\_MD.nii.gz $path/TBSS/${subj}\_FA.nii.gz

done

# TD

subj\_TD=`cat /home/ld/DTI\_YI/Scripts/files/subj\_TD\_DTI.txt`

for subj in $subj\_TD

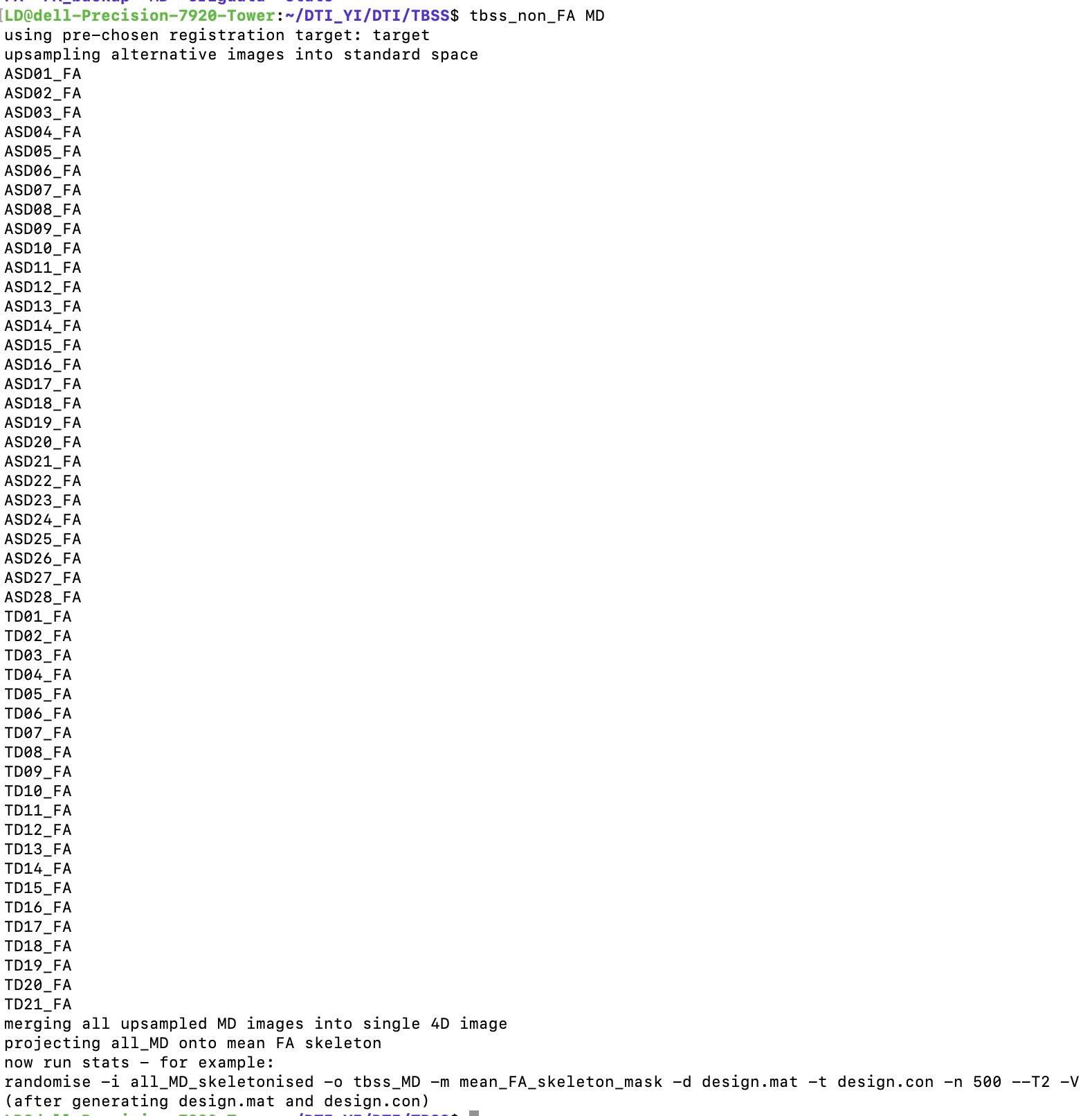
do

cp $path/TD/$subj/dti\_MD.nii.gz $path/TBSS/${subj}\_FA.nii.gz

done

1. 在 TBSS文件夹运行 tbss\_non\_FA

>cd TBSS  
> tbss\_non\_FA MD



结果会在 stats 里生成 all\_MD 和 all\_MD\_skeletonised 文件。

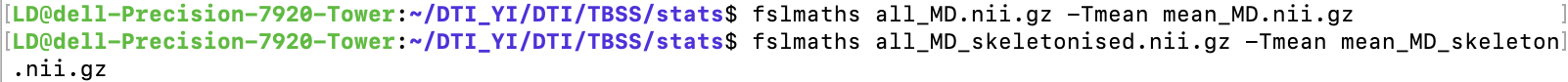
1. 创建 mean\_MD文件

mean\_MD.nii.gz

fslmaths all\_MD.nii.gz -Tmean mean\_MD.nii.gz

mean\_MD\_skeleton.nii.gz ：

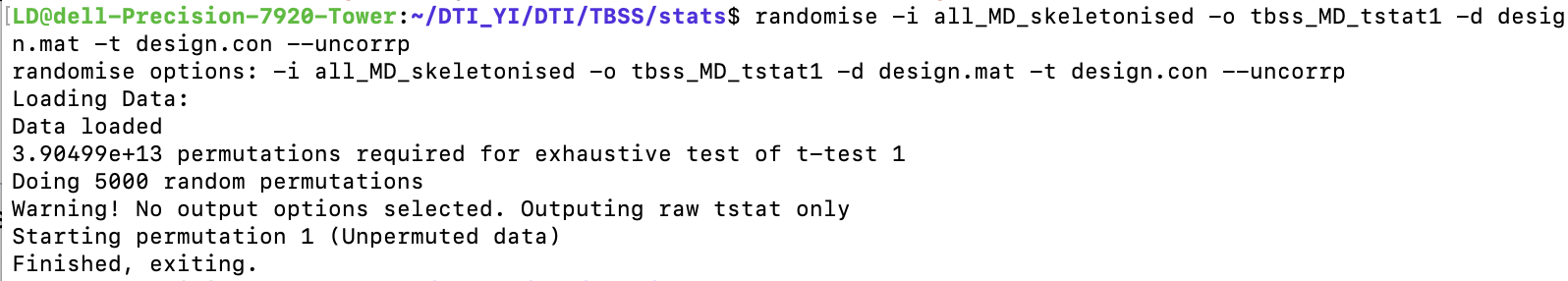
fslmaths all\_MD\_skeletonised.nii.gz -Tmean mean\_MD\_skeleton.nii.gz



1. Randomize, FDR correction, tbss\_fill 与处理 FA 时一致。只需将 FA 文件替换成 MD 文件。

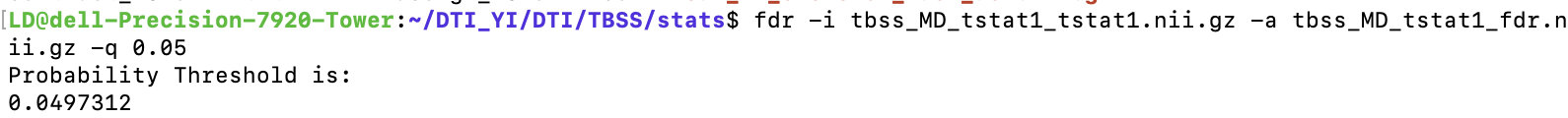
# randomize (for group comparison)

randomise -i all\_MD\_skeletonised -o tbss\_MD\_tstat1 -d design.mat -t design.con --uncorrp



# FDR correction (for group comparison)

fdr -i tbss\_MD\_tstat1\_tstat1.nii.gz -a tbss\_MD\_tstat1\_fdr.nii.gz -q 0.05



# tbss fill (这一步如果无法打开 fsleyes, 可以打开 matlab -nodeskop 后用 DPABI\_VIEW 里查看)

tbss\_fill tbss\_MD\_tstat1\_fdr 0.95 mean\_MD tbss\_fill\_MD\_tstat1\_fdr

fslview mean\_MD -b 0,0.6 mean\_MD\_skeleton -l Green -b 0.2,0.7 tbss\_fill\_MD -l Red-Yellow

# PART 5 相关分析 (FA)

1. Fslsplit & fslmerge

从 all\_FA\_skeletonised.nii.gz中拆分出两组被试的数据：

fslsplit all\_FA\_skeletonised.nii.gz -t

生成所有vol\*.nii.gz 文件，将ASD（前28个）和TD（后21个）分别移至两个文件夹：

**ASD\_FA****\_after\_fslsplit & TD\_FA\_after\_fslsplit**

fslmerge -t ASD28\_FA\_skeletonised.nii.gz ASD\_FA\_after\_fslsplit/\*

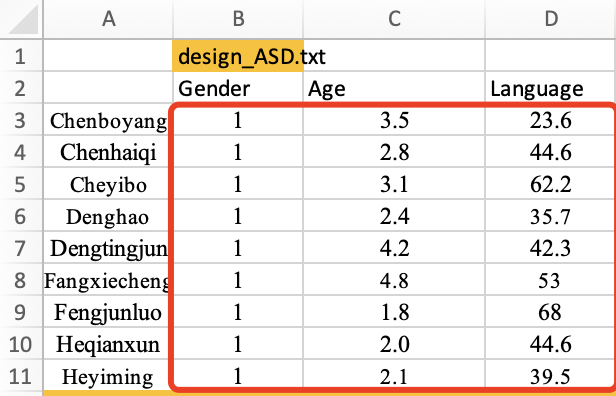
fslmerge -t TD21\_FA\_skeletonised.nii.gz TD\_FA\_after\_fslsplit/\*

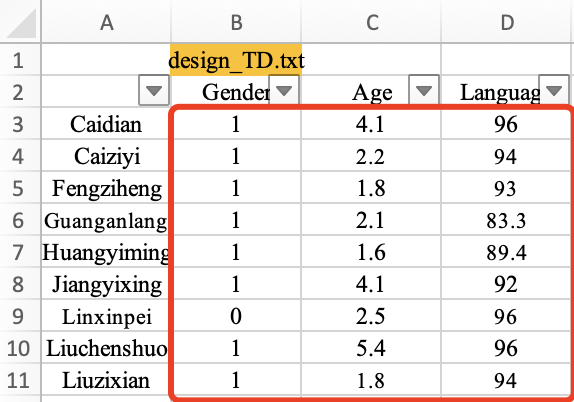


生成两个可用于randomise的文件：

**ASD28\_FA\_skeletonised.nii.gz & TD21\_FA\_skeletonised.nii.gz**

1. 创建两组各自的 matrix文件和共同的 contrast 文件。





A picture containing text, screenshot, line, font

Description automatically generated

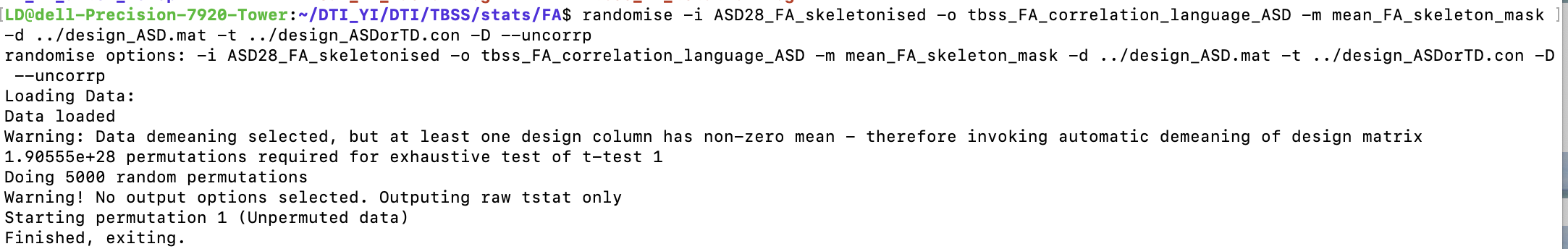
Text2Vest design\_ASD.txt design\_ASD.mat

Text2Vest design\_TD.txt design\_TD.mat

Text2Vest contrast\_ASDorTD.txt design\_ASDorTD.con

1. randomise
2. ASD correlation analysis: with language

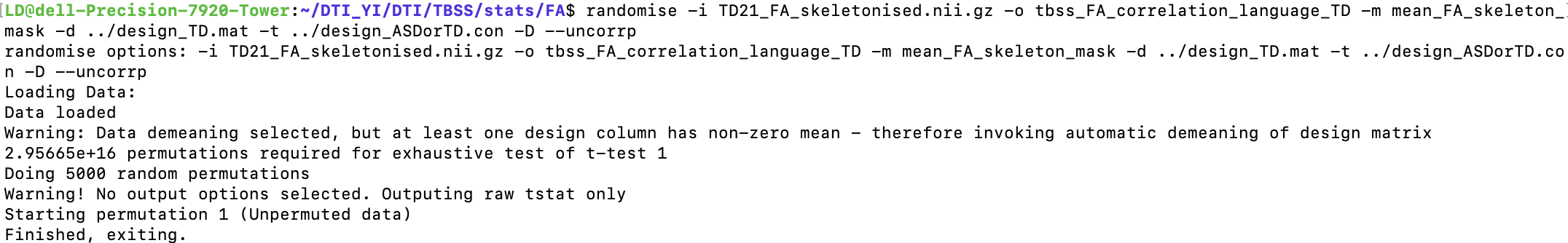
randomise -i ASD28\_FA\_skeletonised -o tbss\_FA\_correlation\_language\_ASD -m mean\_FA\_skeleton\_mask -d ../design\_ASD.mat -t ../design\_ASDorTD.con -D --uncorrp



生成tbss\_FA\_correlation\_language\_ASD\_tstat1.nii.gz

2) TD correlation analysis: with language

randomise -i TD21\_FA\_skeletonised.nii.gz -o tbss\_FA\_correlation\_language\_TD -m mean\_FA\_skeleton\_mask -d ../design\_TD.mat -t ../design\_ASDorTD.con -D --uncorrp



生成tbss\_FA\_correlation\_language\_TD\_tstat1.nii.gz

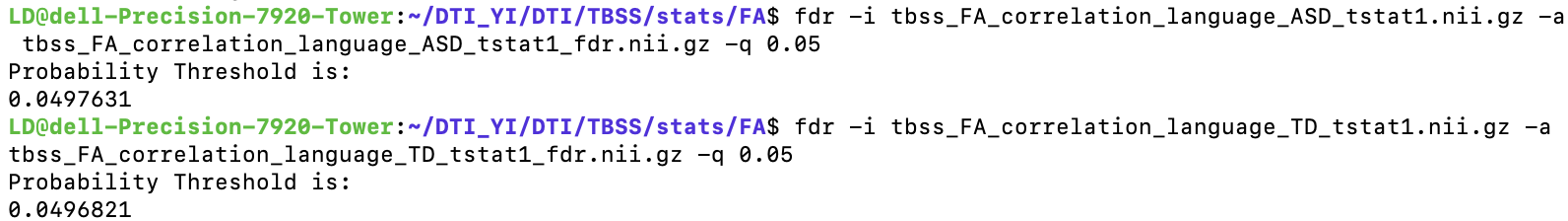
1. FDR correction

ASD:

fdr -i tbss\_FA\_correlation\_language\_ASD\_tstat1.nii.gz -a tbss\_FA\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz -q 0.05

TD:

fdr -i tbss\_FA\_correlation\_language\_TD\_tstat1.nii.gz -a tbss\_FA\_correlation\_language\_TD\_tstat1\_fdr.nii.gz -q 0.05



1. tbss\_fill

tbss\_fill tbss\_FA\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz 0.95 mean\_FA tbss\_fill\_tbss\_FA\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz

tbss\_fill tbss\_FA\_correlation\_language\_TD\_tstat1\_fdr.nii.gz 0.95 mean\_FA tbss\_fill\_tbss\_FA\_correlation\_language\_TD\_tstat1\_fdr.nii.gz

A screenshot of a computer

Description automatically generated

fslview mean\_FA -b 0,0.6 mean\_FA\_skeleton -l Green -b 0.2,0.7 tbss\_fill\_tbss\_FA\_correlation\_language\_TD\_tstat1\_fdr.nii.gz -l Red-Yellow

# PART 6 相关分析 (MD)

1. Fslsplit & fslmerge

从 all\_MD\_skeletonised.nii.gz中拆分出两组被试的数据：

fslsplit all\_MD\_skeletonised.nii.gz -t

生成所有vol\*.nii.gz 文件，将ASD（前28个）和TD（后21个）分别移至两个文件夹：

**ASD\_MD\_after\_fslsplit & TD\_MD\_after\_fslsplit**

fslmerge -t ASD28\_MD\_skeletonised.nii.gz ASD\_MD\_after\_fslsplit/\*

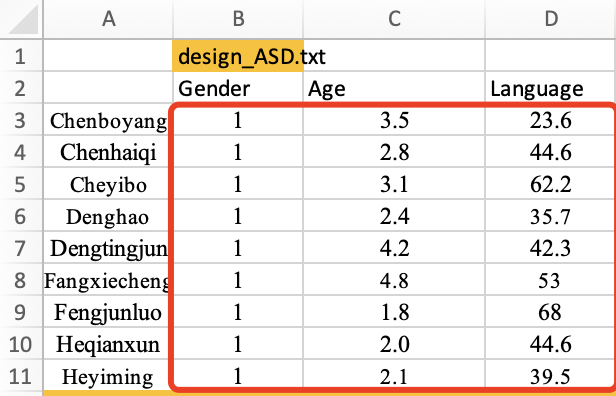
fslmerge -t TD21\_MD\_skeletonised.nii.gz TD\_MD\_after\_fslsplit/\*

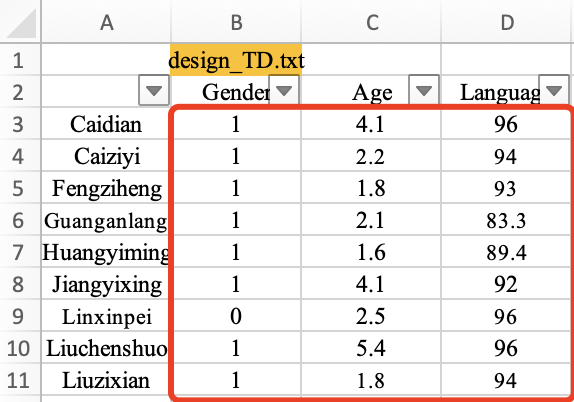


生成两个可用于randomise的文件：

**ASD28\_MD\_skeletonised.nii.gz & TD21\_MD\_skeletonised.nii.gz**

1. 创建两组各自的 matrix文件和共同的 contrast 文件。





A picture containing text, screenshot, line, font

Description automatically generated

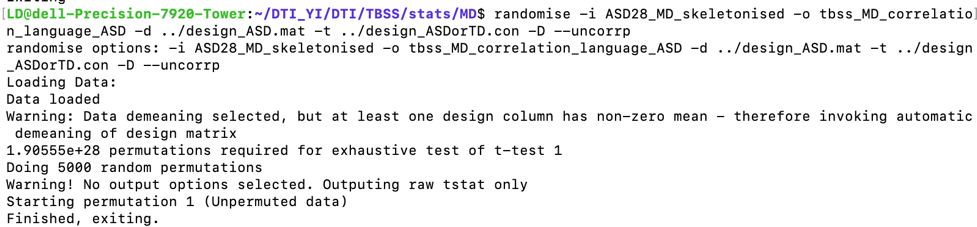
Text2Vest design\_ASD.txt design\_ASD.mat

Text2Vest design\_TD.txt design\_TD.mat

Text2Vest contrast\_ASDorTD.txt design\_ASDorTD.con

1. randomise
2. ASD correlation analysis: with language

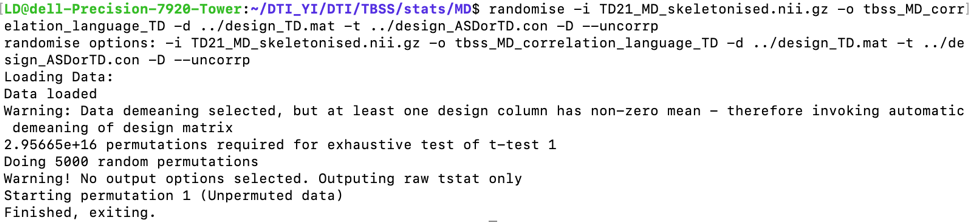
randomise -i ASD28\_MD\_skeletonised -o tbss\_MD\_correlation\_language\_ASD -d ../design\_ASD.mat -t ../design\_ASDorTD.con -D --uncorrp



生成tbss\_MD\_correlation\_language\_ASD\_tstat1.nii.gz

2) TD correlation analysis: with language

randomise -i TD21\_MD\_skeletonised.nii.gz -o tbss\_MD\_correlation\_language\_TD -d ../design\_TD.mat -t ../design\_ASDorTD.con -D --uncorrp

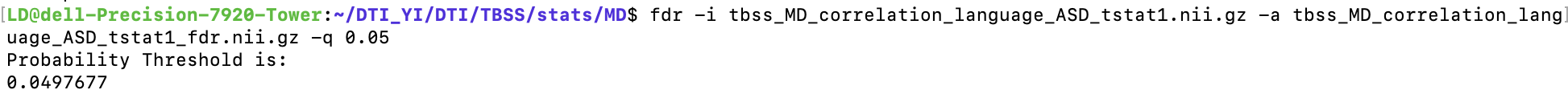


生成tbss\_MD\_correlation\_language\_TD\_tstat1.nii.gz

1. FDR correction

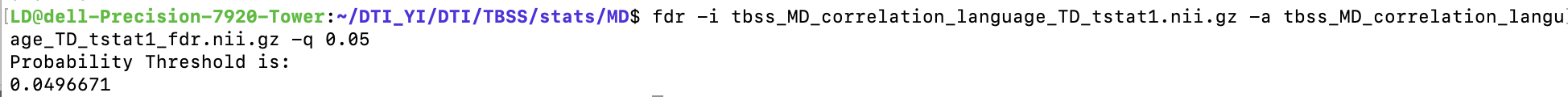
ASD:

fdr -i tbss\_MD\_correlation\_language\_ASD\_tstat1.nii.gz -a tbss\_MD\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz -q 0.05



TD:

fdr -i tbss\_MD\_correlation\_language\_TD\_tstat1.nii.gz -a tbss\_MD\_correlation\_language\_TD\_tstat1\_fdr.nii.gz -q 0.05

****

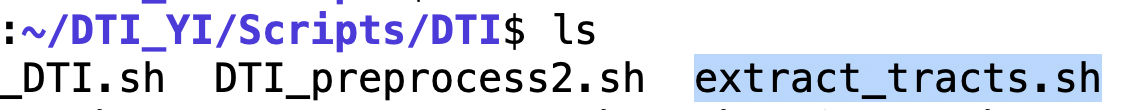
1. tbss\_fill

tbss\_fill tbss\_MD\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz 0.95 mean\_MD tbss\_fill\_tbss\_FA\_correlation\_language\_ASD\_tstat1\_fdr.nii.gz

tbss\_fill tbss\_MD\_correlation\_language\_TD\_tstat1\_fdr.nii.gz 0.95 mean\_MD tbss\_fill\_tbss\_MD\_correlation\_language\_TD\_tstat1\_fdr.nii.gz

fslview mean\_MD -b 0,0.6 mean\_MD\_skeleton -l Green -b 0.2,0.7 tbss\_fill\_tbss\_MD\_correlation\_language\_TD\_tstat1\_fdr.nii.gz -l Red-Yellow

# PART 7 提取感兴趣纤维束（ROIs）



#!/bin/bash

# Extract tracks of interest

path=/home/ld/DTI\_YI/DTI

cd $path

ROIs='41 42 45 46 47 48' # 42 45 46 47 48

for ROI in $ROIs

do

fslmaths mask/JHU-ICBM-labels-1mm.nii.gz -thr $ROI -uthr $ROI mask/JHU\_label$ROI #extract mask

fslmeants -i TBSS/stats/MD/all\_MD\_skeletonised.nii.gz -m mask/JHU\_label$ROI -o TBSS/stats/MD/ind\_MD\_JHU$ROI

done

生成ind\_MD\_JHU41 ind\_MD\_JHU42 …… ind\_MD\_JHU48 这6个文件。

将6个ROIs导出到本地电脑，用R进行组分析（ASD vs TD）和相关分析（与语言的相关）。打开本地电脑终端，输入：（例如）

scp -P …… -r \*\*@\*\*\*.\*\*\*.info:/path/DTI/TBSS/MD