

# 解析TBSS脚本

Alex / 2022-02-01 / [free\\_learner@163.com](mailto:free_learner@163.com) / [learning-archive.org](http://learning-archive.org)

更新于2023-09-19，主要是文字排版上的更新，内容基本保持不变。

学习TBSS实现细节并根据自己的需求进行修改。

## 一、背景

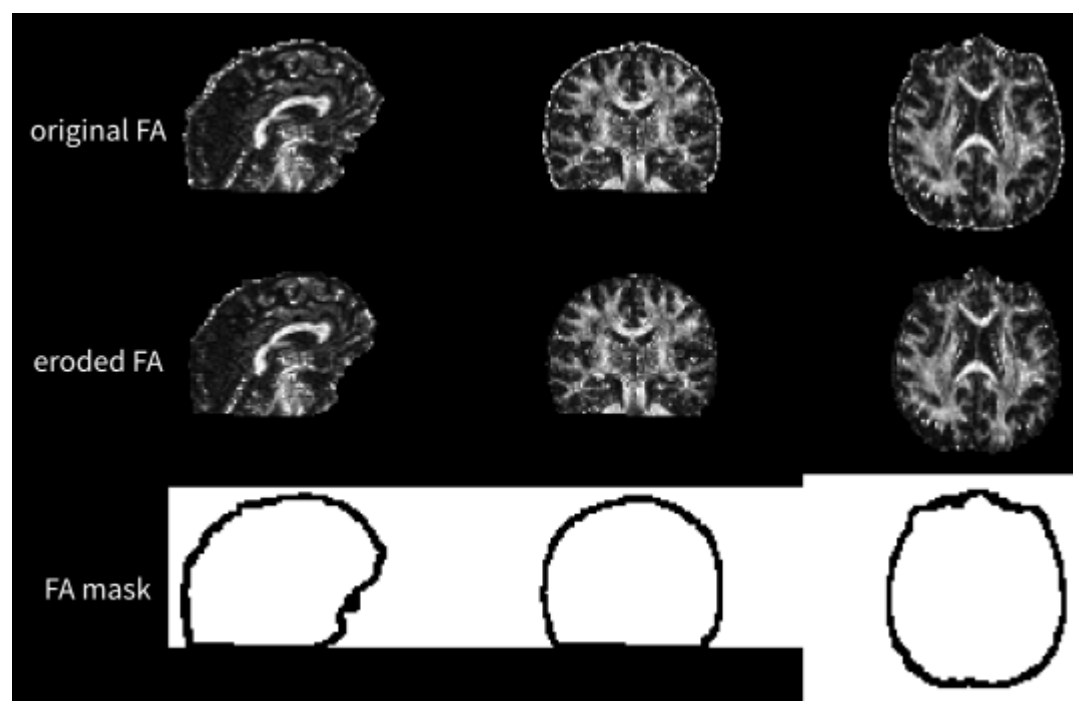
TBSS (Tract-Based Spatial Statistics) 是一种对DTI指标进行组水平分析的方法，目的在于解决被试间空间对应 (spatial correspondence) 的问题。如何使用TBSS脚本进行分析，可以参考我以前的[博客](#)。TBSS脚本对目录结构或文件命名等有一些要求，使用起来不是很方便，因此学习一下代码细节并根据自己的需求修改脚本。

## 二、TBSS脚本

TBSS脚本主要包含5个Bash脚本：`tbss_1_preproc` 用于对FA图像进行一些预处理；`tbss_2_reg` 用于FA图像的配准；`tbss_3_postreg` 用于生成FA骨架；`tbss_4_prestats` 将FA图像投射到FA骨架上；`tbss_non_FA` 将其他指标（比如MD）投射到FA骨架上。下面分别解析这几个脚本的核心功能（对于参数解析等不再赘述）。我这里使用的FSL版本是5.0.11。

### 1. `tbss_1_preproc`

```
## $@表示每个被试的FA图像，for循环表示对每个被试进行相同的操作。
for f in `${FSLDIR}/bin/imglob $@` ; do
    ## 使用fslval计算FA图像的维度，并减去2。
    X=`${FSLDIR}/bin/fslval $f dim1`; X=`echo "$X 2 - p" | dc -`
    Y=`${FSLDIR}/bin/fslval $f dim2`; Y=`echo "$Y 2 - p" | dc -`
    Z=`${FSLDIR}/bin/fslval $f dim3`; Z=`echo "$Z 2 - p" | dc -`
    ## -min 1表示如果FA>1，则使FA=1；
    ## -ero表示对图像边界进行一点收缩（将一些边缘位置的体素设置为0）；
    ## -roi表示将图像第一层和最后一层设置为0。
    ${FSLDIR}/bin/fslmaths $f -min 1 -ero -roi 1 $X 1 $Y 1 $Z 0 1 FA/${f}_FA
    ## 根据FA>0进行二值化。
    ${FSLDIR}/bin/fslmaths FA/${f}_FA -bin FA/${f}_FA_mask
    ## 将FA mask边缘进行扩张（将一些边缘位置的体素设置为1）；
    ## -sub 1 -abs -add FA_mask表示去除扩张部分体素后的mask；
    ## 这个mask用于后续配准中，作用我猜测是为了避免FA图像边缘部分对配准造成影响。
    ${FSLDIR}/bin/fslmaths FA/${f}_FA_mask -dild -dild -sub 1 -abs -add FA/${f}_FA_mask
    FA/${f}_FA_mask -odt char
done
```



2. tbss\_2\_reg

```

## 定义一个函数用于配准。
do_reg(){
    ## 第一个参数表示配准的目标文件，该文件可以是FMRIB58_FA_1mm（-T选项），或者所有被试里最佳的FA图像（-n选项），或者用户指定的输入文件（-t选项）。
    f=$1
    ## 将每个被试的FA图像配准到目标文件上。
    for g in ` $FSLDIR/bin/imglob *_FA.* ` ; do
        o=${g}_to_$f
        ## 使用fsl_reg进行配准，fsl_reg本身也是一个Bash脚本，在下一节单独讨论；
        ## 将命令输出到一个文件中（.commands），配合fsl_sub可以实现并行处理（如果有SGE系统的话）。由于这部分与TBSS核心处理无关，fsl_sub脚本的内容不深入讨论。
        echo "$FSLDIR/bin/fsl_reg $g $f ${g}_to_$f -e -FA" >> .commands
    done
}
## 如果选择最佳FA图像作为配准的目标文件。
if [ $1 = -n ] ; then
    ## 将每个被试的FA图像都当做目标文件进行配准。
    for f in ` $FSLDIR/bin/imglob *_FA.* ` ; do
        do_reg $f
    done
else
    ## 将FMRIB58_FA_1mm或输入文件作为配准的目标文件。
    if [ $1 = -T ] ; then
        TARGET=$FSLDIR/data/standard/FMRIB58_FA_1mm
    elif [ $1 = -t ] ; then
        TARGET=$2
    fi
    ## 将配准的目标文件复制到输出文件夹并重命名为target.nii.gz。
    $FSLDIR/bin/imcp $TARGET FA/target
    do_reg target
fi

```

### 3. fsl\_reg

## 前三个参数分别为配准的源文件（相对于目标文件而言），配准的目标文件和输出文件的前缀；  
## 在TBSS的情况下，配准的源文件就是每个被试的FA图像。

```
INPUT=`${FSLDIR}/bin/remove_ext $1`
```

```
REFERENCE=`${FSLDIR}/bin/remove_ext $2`
```

```
I2R=$3
```

```
shift 3
```

## 解析其他配准参数，默认配准是用 T1\_2\_MNI152\_2mm参数文件（该文件位于

`${FSLDIR}/etc/flirtsch/`），-FA选项表示使用 FA\_2\_FMRIB58\_1mm参数文件；

## -e选项表示只生成配准文件，而不对图像应用配准文件（即对图像进行空间变换）；

## 此外，还可以设置flirt/fnirt的选项以及是否只做仿射配准。

```
apply=1
```

```
affineonly=0
```

```
config="--config=T1_2_MNI152_2mm"
```

```
while [ $_1 != _ ] ; do
```

```
  if [ $1 = -e ] ; then
```

```
    apply=0
```

```
    shift
```

```
  elif [ $1 = -a ] ; then
```

```
    affineonly=1
```

```
    shift
```

```
  elif [ $1 = -FA ] ; then
```

```
    config="--config=FA_2_FMRIB58_1mm"
```

```
    shift
```

```
  elif [ $1 = -flirt ] ; then
```

```
    flirtopts=$2
```

```
    shift 2
```

```
  elif [ $1 = -fnirt ] ; then
```

```
    fnirtopts=$2
```

```
    shift 2
```

```
  fi
```

```
done
```

## 如果检测到存在配准的源文件对应的mask文件，则将该mask文件作为flirt的权重参数。

```
if [ `${FSLDIR}/bin/imtest ${INPUT}_mask` = 1 ] ; then
```

```
  INMASK="-inweight ${INPUT}_mask"
```

```
fi
```

## 使用flirt进行仿射配准。

```
${FSLDIR}/bin/flirt -ref $REFERENCE -in $INPUT $INMASK -omat ${I2R}.mat $flirtopts
```

## 只做仿射配准，而且将配准结果应用到源文件上。

```
if [ $affineonly = 1 ] ; then
```

```
  if [ $apply = 1 ] ; then
```

```
    ${FSLDIR}/bin/flirt -ref $REFERENCE -in $INPUT -applyxfm -init ${I2R}.mat -out $I2R
```

```
$flirtopts
```

```
  fi
```

```
else
```

## 使用fnirt进行非线性配准，并将仿射配准的结果作为输入。

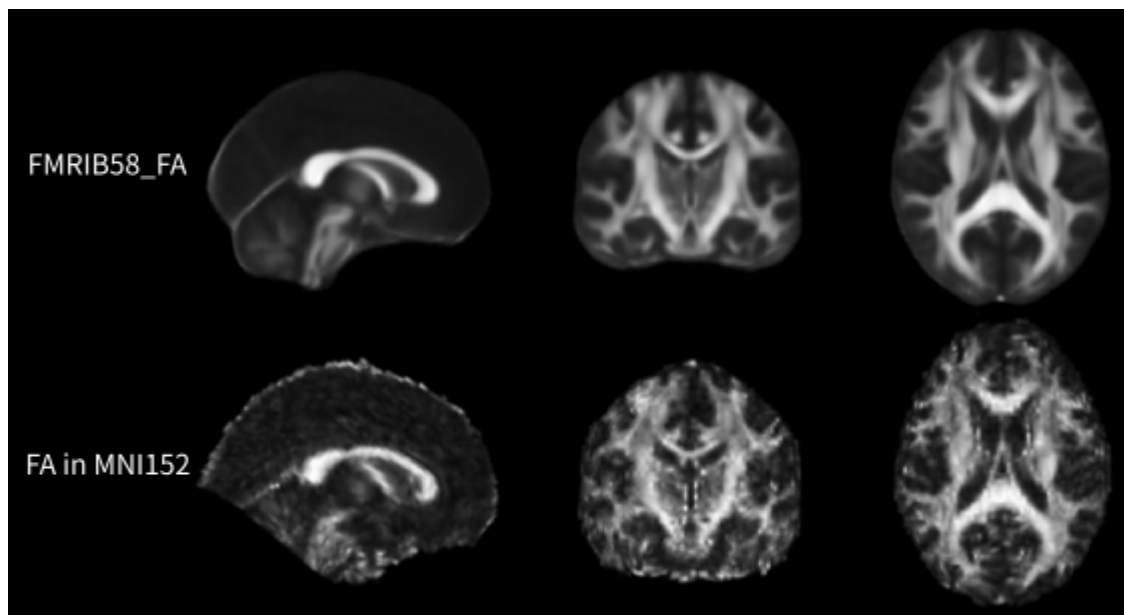
```
${FSLDIR}/bin/fnirt --in=$INPUT --aff=${I2R}.mat --cout=${I2R}_warp $config --
```

```
ref=$REFERENCE $fnirtopts
```

```

## 是否将非线性配准的结果应用到源文件上。
if [ $apply = 1 ] ; then
    ${FSLDIR}/bin/applywarp -i $INPUT -o $I2R -r $REFERENCE -w ${I2R}_warp
fi
## 估计平均（中位数）变形程度，在TBSS中用于选择最佳目标文件。
${FSLDIR}/bin/fslmaths ${I2R}_warp -sqr -Tmean ${I2R}_tmp
${FSLDIR}/bin/fslstats ${I2R}_tmp -M -P 50 > ${I2R}_warp.msf
fi

```



#### 4. tbss\_3\_postreg

```

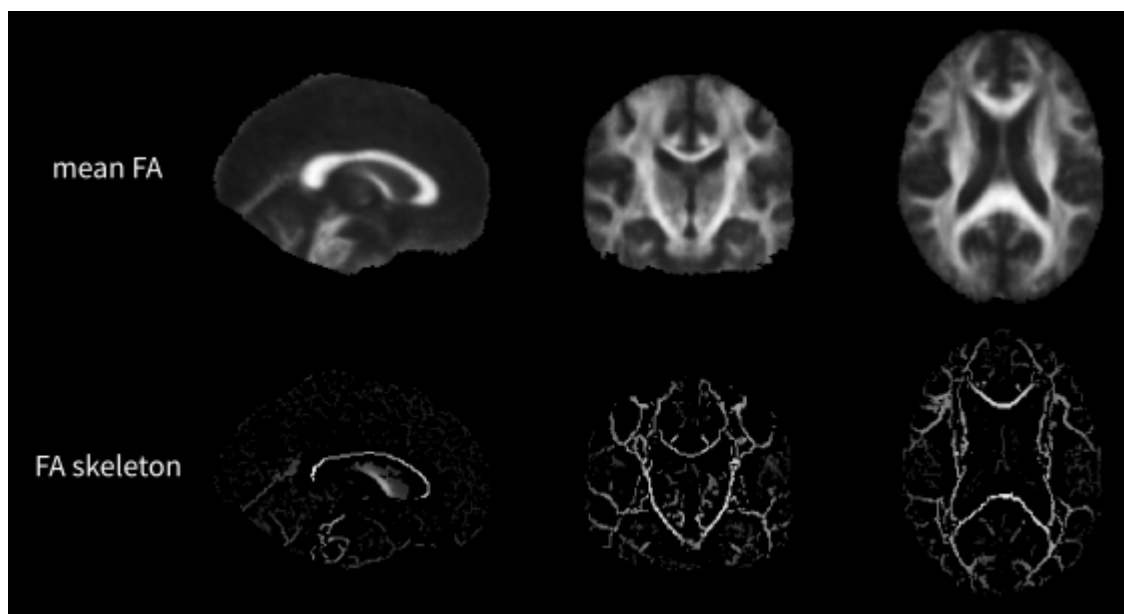
## -S选项表示根据当前研究中所有被试生成FA骨架，-T选项表示用FMRIB58_FA对应的骨架。
if [ $1 = -S ] ; then
    SKELETON=0
elif [ $1 = -T ] ; then
    SKELETON=1
fi
## 判断是否存在target.nii.gz文件，如果不存在说明需要从所有被试的FA图像选择最佳图像作为配准
的目标文件。
if [ `${FSLDIR}/bin/imtest target` = 1 ] ; then
    best=target
else
    ## 计算被试量。
    n=`$FSLDIR/bin/imglob *_FA.nii* *_FA.img* *_FA.hdr* | wc -w`
    ## 计算每个被试与其他所有被试的配准的平均（中位数）变形程度。
    for f in `${FSLDIR}/bin/imglob *_FA.nii* *_FA.img* *_FA.hdr*` ; do
        meanscore=0
        medianscore=0
        for g in `${FSLDIR}/bin/imglob *_FA.nii* *_FA.img* *_FA.hdr*` ; do
            thismeanscore=`cat ${g}_to_${f}_warp.msf | awk '{print $1}'`
            thismedianscore=`cat ${g}_to_${f}_warp.msf | awk '{print $2}'`
            meanscore=`echo "10 k $meanscore $thismeanscore + p" | dc -`
            medianscore=`echo "10 k $medianscore $thismedianscore + p" | dc -`
        done
        meanscore=`echo "10 k $meanscore $n / p" | dc -`
        medianscore=`echo "10 k $medianscore $n / p" | dc -`
        echo "$f $meanscore $medianscore"
        echo "$f $meanscore $medianscore" >> all.msf
    done
    ## 选择平均变形程度最小的被试图像作为配准的目标文件；all.msf第1列是被试图像名字，第2列是
    平均变形程度。
    best=`cat all.msf | sort -k 2 -n | head -n 1 | awk '{print $1}'`
    $FSLDIR/bin/imcp $best target
fi
## 根据文件维度来判断配准目标文件是否位于MNI152空间
if [ `${FSLDIR}/bin/imtest target` = 1 ] && [ `${FSLDIR}/bin/fslval $best dim1` = 182 ] && [
`${FSLDIR}/bin/fslval $best dim2` = 218 ] && [ `${FSLDIR}/bin/fslval $best dim3` = 182 ] ;
then
    ## 将每个被试的FA配准到MNI152空间。
    filelist=`$FSLDIR/bin/imglob *_FA.nii* *_FA.img* *_FA.hdr*`;
    for f in $filelist ; do
        echo $f
        $FSLDIR/bin/applywarp -i $f -o ${f}_to_target -r target -w ${f}_to_target_warp --rel
    done
else
    ## 如果目标文件不在MNI152空间，将目标文件仿射变换到MNI152空间。
    $FSLDIR/bin/fslmaths target -bin target_mask
    $FSLDIR/bin/flirt -in target -inweight target_mask -ref

```

```

$FSLDIR/data/standard/FMRIB58_FA_1mm -omat target_to_MNI152.mat -out target_to_MNI152
## 将每个被试的FA配准到MNI152空间。
for f in ` $FSLDIR/bin/imglob *_FA.nii* *_FA.img* *_FA.hdr* ` ; do
    $FSLDIR/bin/applywarp -i $f -o ${f}_to_target -r $FSLDIR/data/standard/FMRIB58_FA_1mm
-w ${f}_to_${best}_warp --postmat=target_to_MNI152.mat
done
fi
## 将所有被试的位于MNI152空间的FA文件合并起来。
${FSLDIR}/bin/fslmerge -t ../stats/all_FA ` $FSLDIR/bin/imglob *_FA_to_target.* `
## 计算所有被试的平均FA。
$FSLDIR/bin/fslmaths all_FA -max 0 -Tmin -bin mean_FA_mask -odt char
$FSLDIR/bin/fslmaths all_FA -mas mean_FA_mask all_FA
$FSLDIR/bin/fslmaths all_FA -Tmean mean_FA
## 生成FA骨架。
if [ $SKELETON = 0 ] ; then
    ## 根据所有被试平均FA生成骨架。
    $FSLDIR/bin/tbss_skeleton -i mean_FA -o mean_FA_skeleton
else
    ## 使用 FMRIB58_FA的骨架。
    $FSLDIR/bin/fslmaths $FSLDIR/data/standard/FMRIB58_FA_1mm -mas mean_FA_mask mean_FA
    $FSLDIR/bin/fslmaths mean_FA -bin mean_FA_mask
    $FSLDIR/bin/fslmaths all_FA -mas mean_FA_mask all_FA
    $FSLDIR/bin/imcp $FSLDIR/data/standard/FMRIB58_FA-skeleton_1mm mean_FA_skeleton
fi

```



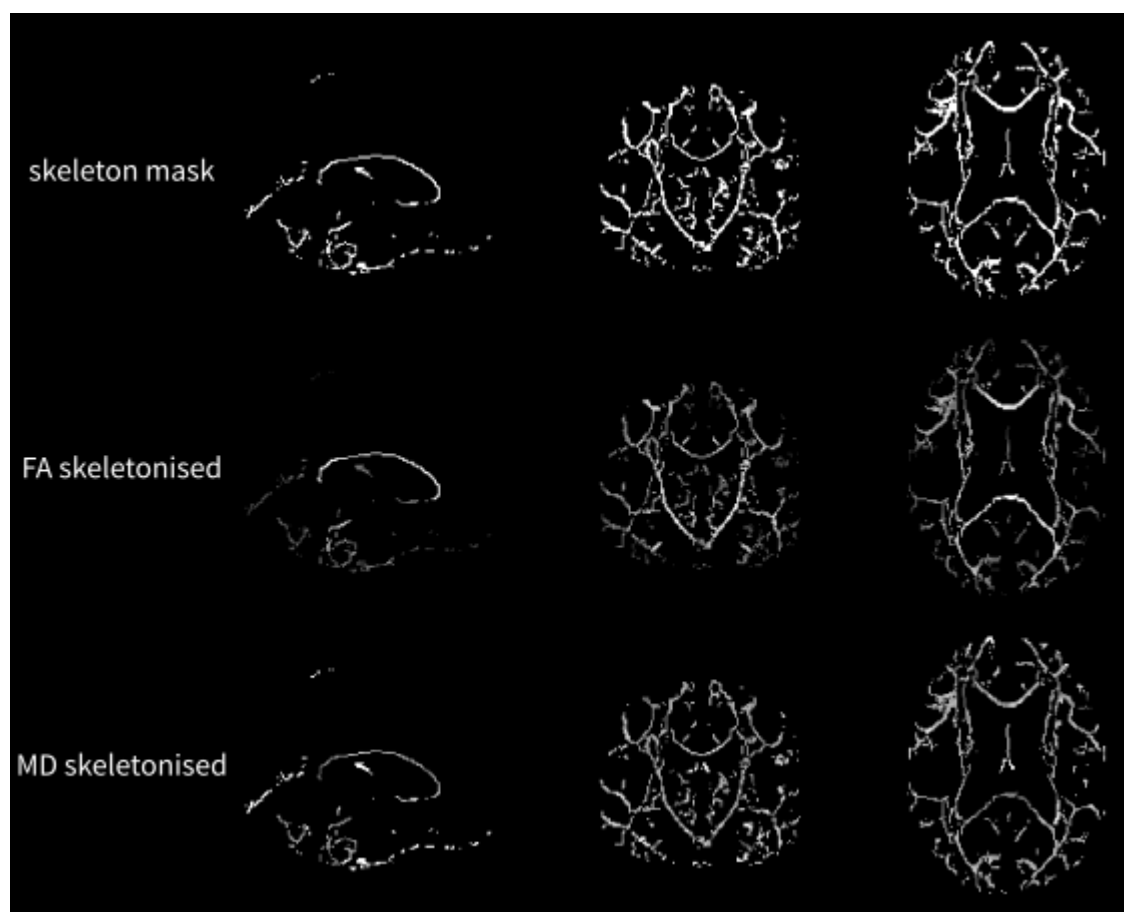
## 5. tbss\_4\_prestats

```
## 根据阈值（比如，FA=0.2）生成骨架mask。
thresh=$1
${FSLDIR}/bin/fslmaths mean_FA_skeleton -thr $thresh -bin mean_FA_skeleton_mask
## 生成骨架距离图像并将被试FA图像投射到骨架上。
${FSLDIR}/bin/fslmaths mean_FA_mask -mul -1 -add 1 -add mean_FA_skeleton_mask
mean_FA_skeleton_mask_dst
${FSLDIR}/bin/distancemap -i mean_FA_skeleton_mask_dst -o mean_FA_skeleton_mask_dst
${FSLDIR}/bin/tbss_skeleton -i mean_FA -p $thresh mean_FA_skeleton_mask_dst
${FSLDIR}/data/standard/LowerCingulum_1mm all_FA all_FA_skeletonised
```

## 6. tbss\_non\_FA

```
## 将每个被试的其他指标（比如，MD）投射到MNI152空间。
ALTIM=$1
if [ -f target_to_MNI152.mat ] ; then
    postaffine="--postmat=target_to_MNI152.mat"
fi
for f in ` ${FSLDIR}/bin/imglob *_FA.* ` ; do
    ## 要求其他指标的文件名和FA图像的文件名一样。
    f=` echo $f | sed 's/_FA$/g' `
    echo $f
    ${FSLDIR}/bin/applywarp -i ../${ALTIM}/${f} -o ${f}_to_target_${ALTIM} -r
    ${FSLDIR}/data/standard/FMRIB58_FA_1mm -w ${f}_FA_to_${best}_warp $postaffine
done
## 将所有被试的其他指标合并起来。
${FSLDIR}/bin/fslmerge -t ../stats/all_${ALTIM} ` ${FSLDIR}/bin/imglob *_to_target_${ALTIM}.* `
${FSLDIR}/bin/fslmaths all_${ALTIM} -mas mean_FA_mask all_${ALTIM}
## 将所有被试的其他指标投射到FA骨架上。
thresh=`cat thresh.txt`
${FSLDIR}/bin/tbss_skeleton -i mean_FA -p $thresh mean_FA_skeleton_mask_dst
${FSLDIR}/data/standard/LowerCingulum_1mm all_FA all_${ALTIM}_skeletonised -a all_${ALTIM}
```





### 三、修改脚本

---

在日常的使用中，我通过只是使用固定的参数，因此可以简化TBSS的脚本，同时使得在目录结构和文件命名上更灵活。修改后的脚本如下，通过测试结果和直接使用TBSS脚本一致。

```

#!/bin/bash
## a text file contains the absolute paths of all FA images
FAlist=$1
## output folder to save results
OUTDIR=$2
mkdir -p ${OUTDIR}
## the number of subjects
NSUB=$(cat $FAlist | wc -l)
## -----
## Step1: erode FA and make FA mask for each subject
FADIR=${OUTDIR}/FA
mkdir -p ${FADIR}
for idx in $(seq 1 $NSUB)
do
    ## padding with zeros, assuming less than 1000 subjects
    SUBID=$(printf "%03d" $idx)
    FAIMG=$(cat $FAlist | sed -n "${idx}"p)
    ## erode a little and zero end slices
    X=$(fslval $FAIMG dim1)
    X=$(( X - 2 ))
    Y=$(fslval $FAIMG dim2)
    Y=$(( Y - 2 ))
    Z=$(fslval $FAIMG dim3)
    Z=$(( Z - 2 ))
    fslmaths $FAIMG -min 1 -ero -roi 1 $X 1 $Y 1 $Z 0 1 ${FADIR}/${SUBID}.nii.gz
    ## create mask (for use in FLIRT & FNIRT)
    fslmaths ${FADIR}/${SUBID}.nii.gz -bin ${FADIR}/${SUBID}_mask.nii.gz
    fslmaths ${FADIR}/${SUBID}_mask.nii.gz -dild -dilD -sub 1 -abs -add ${FADIR}/${SUBID}_mask
    ${FADIR}/${SUBID}_mask.nii.gz -odt char
done
## -----
## Step2: register FA into FMRIB58_FA template
REGDIR=${OUTDIR}/REG
mkdir -p ${REGDIR}
REFERENCE=${FSLDIR}/data/standard/FMRIB58_FA_1mm.nii.gz
for idx in $(seq 1 $NSUB)
do
    SUBID=$(printf "%03d" $idx)
    flirt -ref $REFERENCE -in ${FADIR}/${SUBID}.nii.gz -inweight ${FADIR}/${SUBID}_mask.nii.gz -
    omat ${REGDIR}/${SUBID}toMNI.mat
    fnirt --in=${FADIR}/${SUBID}.nii.gz --aff=${REGDIR}/${SUBID}toMNI.mat --
    cout=${REGDIR}/${SUBID}toMNI_warp --config=FA_2_FMRIB58_1mm --ref=$REFERENCE
    applywarp -i ${FADIR}/${SUBID}.nii.gz -o ${REGDIR}/${SUBID}toMNI.nii.gz -r $REFERENCE -w
    ${REGDIR}/${SUBID}toMNI_warp --rel
done
## -----
## Step3: create FA skeleton based on mean FA of all subjects

```

```

STATDIR=${OUTDIR}/STATS
mkdir -p ${STATDIR}
THR=0.2
fslmerge -t ${STATDIR}/all_FA.nii.gz $(ls ${REGDIR}/*toMNI.nii.gz)
fslmaths ${STATDIR}/all_FA.nii.gz -max 0 -Tmin -bin ${STATDIR}/mean_FA_mask.nii.gz -odt char
fslmaths ${STATDIR}/all_FA.nii.gz -mas ${STATDIR}/mean_FA_mask.nii.gz ${STATDIR}/all_FA.nii.gz
fslmaths ${STATDIR}/all_FA.nii.gz -Tmean ${STATDIR}/mean_FA.nii.gz
tbss_skeleton -i ${STATDIR}/mean_FA.nii.gz -o ${STATDIR}/mean_FA_skeleton.nii.gz
fslmaths ${STATDIR}/mean_FA_skeleton.nii.gz -thr $THR -bin
${STATDIR}/mean_FA_skeleton_mask.nii.gz
## -----
## Step4: project FA of each subject onto the skeleton
fslmaths ${STATDIR}/mean_FA_mask.nii.gz -mul -1 -add 1 -add
${STATDIR}/mean_FA_skeleton_mask.nii.gz ${STATDIR}/mean_FA_skeleton_mask_dst.nii.gz
distancemap -i ${STATDIR}/mean_FA_skeleton_mask_dst.nii.gz -o
${STATDIR}/mean_FA_skeleton_mask_dst.nii.gz
tbss_skeleton -i ${STATDIR}/mean_FA.nii.gz -p $THR ${STATDIR}/mean_FA_skeleton_mask_dst.nii.gz
${FSLDIR}/data/standard/LowerCingulum_1mm ${STATDIR}/all_FA.nii.gz
${STATDIR}/all_FA_skeletonised.nii.gz

```