

# 解析FSL-VBM脚本

Alex / 2022-08-01 / [free\\_learner@163.com](mailto:free_learner@163.com) / [AlexBrain.cn](http://AlexBrain.cn)

更新于2023-09-22，主要是文字排版上的更新，内容基本保持不变。

## 一、背景

在以前的[博客](#)里，我尝试用FSL进行VBM分析（发现效果很差），这里学习一下FSL-VBM脚本实现细节，目的是加深对于图像处理的相关步骤的理解。

## 二、脚本结构

FSL-VBM[流程](#)主要包含三个脚本：`fslvbm_1_bet`，`fslvbm_2_template` 和 `fslvbm_3_proc`，我学习的方式是对每个脚本添加注释。

## 三、具体内容

### 1. fslvbm\_1\_bet

```
## 定义一个Usage函数，用于显示脚本用法
Usage() {
    echo ""
    echo "Usage: fslvbm_1_bet <-b or -N> [options to be passed on to BET] "
    echo ""
    echo "-b : default BET brain extraction (which uses -f 0.4)"
    echo "-N : increased robustness in the brain extraction when a lot of neck is present"
    echo ""
    echo "e.g.,  fslvbm_1_bet -b -f 0.6 -R"
    echo ""
    exit 1
}
```

```

## 如果没有加任何选项，运行Usage，显示脚本用法
[ "$1" = "" ] && Usage
## 将日期、主机名字、系统信息、当前路径、脚本名字和所有参数保存到.fslvbmlog文件中
echo [`date`] [`hostname`] [`uname -a`] [`pwd`] [$0 $@] >> .fslvbmlog
## 如果第一个参数是-N，则将bet置1。如果输入图像中包含太多脖子的部分，则需要使用-N选项
bet=0
if [ $1 = -N ] ; then
    bet=1
fi
shift
## 新建名为struc的文件夹，将输入文件（T1加权像）复制到新文件夹中，在原文件名上加上_struc
mkdir -p struc
for g in `FSLDIR/bin/imglob *` ; do
    imcp $g struc/${g}_struc
done
## 对每个被试的T1像进行脑提取，如果bet为0，则使用默认参数（-f 0.4），再加上其他参数（$@）
## 如果bet为1，则去除图像中包含的脖子部分，standard_space_roi也是一个Bash脚本，这里不展开
## 脑提取后的文件在原文件名基础上添加了_brain
cd struc
imagelist=""
for g in `FSLDIR/bin/imglob *_struc.*` ; do
    echo $g
    if [ $bet = 0 ] ; then
        FSLDIR/bin/bet $g ${g}_brain -f 0.4 $@
    else
        FSLDIR/bin/bet $g ${g}_brain
        FSLDIR/bin/standard_space_roi ${g}_brain ${g}_cut -roiNONE -ssref
        {FSLDIR}/data/standard/MNI152_T1_2mm_brain -altinput $g
        FSLDIR/bin/bet ${g}_cut ${g}_brain -f 0.4 $@
    fi
    imagelist="$imagelist $g ${g}_brain"
done
## slicesdir生成一些截面图用于检查脑提取质量，注意其中imagelist生成的方法
echo "Done running BET, now running slicesdir:"
FSLDIR/bin/slicesdir -o $imagelist

```

## 2. fslvbm\_2\_template

```

## 同上脚本，用于显示脚本用法
Usage() {
    echo ""
    echo "Usage: fslvbm_2_template [options]"
    echo ""
    echo "-n : nonlinear registration (recommended)"
    echo "-a : affine registration (discouraged)"
    echo ""
    exit 1
}
## 同上
[ "$1" = "" ] && Usage
## 同上
echo [`date`] [`hostname`] [`uname -a`] [`pwd`] [$0 $@] >> .fslvbmlog
## HOWLONG参数跟后面并行计算有关，但是和VBM分析无关，可以忽略
## 如果-a，则表示用仿射变换生成灰质模板，用变量REG来表示（如果是-n，则REG为空）
HOWLONG=30
if [ $1 = -a ] ; then
    REG="-a"
    HOWLONG=5
fi

cd struc
## T表示灰质概率模板
T=${FSLDIR}/data/standard/tissuepriors/avg152T1_gray
## 对每个脑提取后的文件（_brain）进行组织分割，其中_pve_1表示灰质的部分体积估计（partial volume estimation）
## 这里会使用fsl_sub进行并行计算，如果fsl_sub检测到SGE系统（如果没有，则串行计算），由于和VBM原理无关，这里不展开
### segmentation
/bin/rm -f fslvbm2a
for g in `ls $FSLDIR/bin/imglob *_struc.*` ; do
    echo $g
    echo "$FSLDIR/bin/fast -R 0.3 -H 0.1 ${g}_brain ; \
        $FSLDIR/bin/immv ${g}_brain_pve_1 ${g}_GM" >> fslvbm2a
done
chmod a+x fslvbm2a
fslvbm2a_id=`$FSLDIR/bin/fsl_sub -T 30 -N fslvbm2a -t ./fslvbm2a`
echo Running segmentation: ID=$fslvbm2a_id
## 将每个被试的灰质图仿射配准到灰质模板，fsl_reg也是一个Bash脚本，在以前学习TBSS脚本时有涉及到，这里不重复。
### Estimation of the registration parameters of GM to grey matter standard template
/bin/rm -f fslvbm2b

for g in `ls $FSLDIR/bin/imglob *_struc.*` ; do
    echo "${FSLDIR}/bin/fsl_reg ${g}_GM $T ${g}_GM_to_T -a" >> fslvbm2b
done

```

```

chmod a+x fslvbm2b
fslvbm2b_id=`$FSLDIR/bin/fsl_sub -j $fslvbm2a_id -T $HOWLONG -N fslvbm2b -t ./fslvbm2b`
echo Running initial registration: ID=$fslvbm2b_id

### Creation of the GM template by averaging all (or following the template_list for) the
GM_n1_0 and GM_xflipped_n1_0 images
cat <<stage_tpl3 > fslvbm2c
#!/bin/sh
## 这里是通过template_list或者通过实际存在的文件来获取被试文件名
if [ -f ../template_list ] ; then
    template_list=`cat ../template_list`
    template_list=`$FSLDIR/bin/remove_ext $template_list`
else
    template_list=`echo *_struc.* | sed 's/_struc\.\/\./g'`
    template_list=`$FSLDIR/bin/remove_ext $template_list | sort -u`
    echo "WARNING - study-specific template will be created from ALL input data - may not be
group-size matched!!!"
fi
for g in $template_list ; do
    mergelist="$mergelist ${g}_struc_GM_to_T"
done
## 将所有被试的（仿射变换后的）灰质图平均得到template_GM，然后左右翻转以后再平均，得到左右对称
的初步的灰质模板
$FSLDIR/bin/fslmerge -t template_4D_GM $mergelist
$FSLDIR/bin/fslmaths template_4D_GM -Tmean template_GM
$FSLDIR/bin/fslswapdim template_GM -x y z template_GM_flipped
$FSLDIR/bin/fslmaths template_GM -add template_GM_flipped -div 2 template_GM_init
stage_tpl3
chmod +x fslvbm2c
fslvbm2c_id=`fsl_sub -j $fslvbm2b_id -T 15 -N fslvbm2c ./fslvbm2c`
echo Creating first-pass template: ID=$fslvbm2c_id
## 将所有被试的灰质图仿射或非线性（通过REG来反映）配准到初步的灰质模板上
### Estimation of the registration parameters of GM to grey matter standard template
/bin/rm -f fslvbm2d
T=template_GM_init
for g in `$FSLDIR/bin/imglob *_struc.*` ; do
    echo "${FSLDIR}/bin/fsl_reg ${g}_GM $T ${g}_GM_to_T_init $REG -fnirt \"--
config=GM_2_MNI152GM_2mm.cnf\" >> fslvbm2d
done
chmod a+x fslvbm2d
fslvbm2d_id=`$FSLDIR/bin/fsl_sub -j $fslvbm2c_id -T $HOWLONG -N fslvbm2d -t ./fslvbm2d`
echo Running registration to first-pass template: ID=$fslvbm2d_id
## 将所有被试的配准后的灰质图再次平均、左右翻转、平均，生成最终的灰质模板
### Creation of the GM template by averaging all (or following the template_list for) the
GM_n1_0 and GM_xflipped_n1_0 images
cat <<stage_tpl4 > fslvbm2e
#!/bin/sh
if [ -f ../template_list ] ; then

```

```

template_list=`cat ../template_list`
template_list=`\${FSLDIR}/bin/remove_ext \${template_list}`
else
template_list=`echo *_struc.* | sed 's/_struc\./\./g'`
template_list=`\${FSLDIR}/bin/remove_ext \${template_list} | sort -u`
echo "WARNING - study-specific template will be created from ALL input data - may not be
group-size matched!!!"
fi
for g in \${template_list} ; do
mergelist="\${mergelist} \${g}_struc_GM_to_T_init"
done
\${FSLDIR}/bin/fslmerge -t template_4D_GM \${mergelist}
\${FSLDIR}/bin/fslmaths template_4D_GM -Tmean template_GM
\${FSLDIR}/bin/fslswapdim template_GM -x y z template_GM_flipped
\${FSLDIR}/bin/fslmaths template_GM -add template_GM_flipped -div 2 template_GM
stage_tpl4
chmod +x fslvbm2e
fslvbm2e_id=`fsl_sub -j $fslvbm2d_id -T 15 -N fslvbm2e ./fslvbm2e`
echo Creating second-pass template: ID=$fslvbm2e_id

echo "Study-specific template will be created, when complete, check results with:"
echo "fslview struc/template_4D_GM"
echo "and turn on the movie loop to check all subjects, then run:"
echo "fslview " \${FSLDIR}/data/standard/tissuepriors/avg152T1_gray " struc/template_GM"
echo "to check general alignment of mean GM template vs. original standard space template."

```

### 3. fslvbm\_3\_proc

```

## 同上
echo [`date`] [`hostname`] [`uname -a`] [`pwd`] [$0 $@] >> .fslvbmlog

mkdir -p stats
cd struc

echo "Now running the preprocessing steps and the pre-analyses"
## 将每个被试的灰质图（非线性）配准到上一个脚本得到的灰质模板，并生成Jacobian行列式
## 使用Jacobian行列式来校正（modulate）由于配准引起的（整体和局部的）体积变化，生成_mod文件
/bin/rm -f fslvbm3a
for g in `${FSLDIR}/bin/imglob *_struc.*` ; do
    echo $g
    echo "${FSLDIR}/bin/fsl_reg ${g}_GM template_GM ${g}_GM_to_template_GM -fnirt \\"--
config=GM_2_MNI152GM_2mm.cnf --jout=${g}_JAC_n1\"; \
    ${FSLDIR}/bin/fslmaths ${g}_GM_to_template_GM -mul ${g}_JAC_n1 ${g}_GM_to_template_GM_mod
-odt float" >> fslvbm3a
done
chmod a+x fslvbm3a
fslvbm3a_id=`${FSLDIR}/bin/fsl_sub -T 40 -N fslvbm3a -t ./fslvbm3a`
echo Doing registrations: ID=$fslvbm3a_id

cd ../stats

cat <<stage_preproc2 > fslvbm3b
#!/bin/sh

`${FSLDIR}/bin/imcp ../struc/template_GM template_GM
## 将所有被试的转换到灰质模板空间的校正后的灰质图合并（作为randomise的输入），并生成灰质mask
（阈值0.01）
`${FSLDIR}/bin/fslmerge -t GM_merg \`${FSLDIR}/bin/imglob ../struc/*_GM_to_template_GM.*`\`
`${FSLDIR}/bin/fslmerge -t GM_mod_merg \`${FSLDIR}/bin/imglob
../struc/*_GM_to_template_GM_mod.*`\`

`${FSLDIR}/bin/fslmaths GM_merg -Tmean-thr 0.01 -bin GM_mask -odt char

/bin/cp ../design.* .
## 空间平滑，平滑核的标准差（sigma）为2/3/4mm
## 然后使用randomise进行置换检验
for i in GM_mod_merg ; do
    for j in 2 3 4 ; do
        `${FSLDIR}/bin/fslmaths ${i} -s ${j} ${i}_s${j}
        `${FSLDIR}/bin/randomise -i ${i}_s${j} -o ${i}_s${j} -m GM_mask -d design.mat -t
design.con -V
    done
done

stage_preproc2

```

```
chmod a+x fslvbm3b
```

```
fslvbm3b_id=`${FSLDIR}/bin/fsl_sub -T 15 -N fslvbm3b -j $fslvbm3a_id ./fslvbm3b`
```

```
echo Doing subject concatenation and initial randomise: ID=$fslvbm3b_id
```

```
echo "Once this has finished, run randomise with 5000 permutations on the 'best' smoothed 4D  
GM_mod_merg. We recommend using the -T (TFCE) option. For example:"
```

```
echo "randomise -i GM_mod_merg_s3 -o GM_mod_merg_s3 -m GM_mask -d design.mat -t design.con -n  
5000 -T -V"
```