

使用FSL自动报告脑区结果

Alex / 2020-04-23 / free_learner@163.com / AlexBrain.cn

更新于2023-08-24，主要是文字排版上的更新，内容基本保持不变。

在网上看到一个Bash脚本，可以自动报告脑区结果，这里学习一下实现过程。该脚本来自于中科院的孔亚卓研究员，脚本内容如下并加上了我自己的（中文）注释：

```
#!/bin/sh

# Cluster_atlas - report cluster MNI coordinate, cluster size and atlas
# $1 - filename of input volume, e.g. thresholded stats map, must register to MNI 2mm space
# $2 - threshold for input volume
# $3 - cluster size i.e. minimum voxel number in one cluster
# atlasname reported here
#     1: Harvard-Oxford Cortical Structural Atlas
#     2: Harvard-Oxford Subcortical Structural Atlas
#     3: MNI Structural Atlas
#
# Yazhuo Kong, FMRIB, OXFORD -- 07/04/2009
## 该脚本实现的功能是给定一个阈值和团块大小，报告通过阈值的团块(cluster)的脑区，脑区信息通过FSL
自带的三个Atlas来确定。
if [ $# -ne 3 ] ; then
    echo "Usage: cluster_atlas.sh <input volume> <threshold> <cluster size>"
    echo " <1> input volume, e.g. thresholded stats map"
    echo " <2> T or Z threshold for input volume"
    echo " <3> cluster size i.e. minimum voxel number in one cluster"
    exit 1
fi
## 检查参数个数，使用的是三个位置参数。
# some variables
inputfile=`${FSLDIR}/bin/imglob -oneperimage $1`;
## 去掉文件后缀，比如输入/home/alex/t1.nii.gz返回/home/alex/t1
fullpath=`dirname $1`          # path to file
## 获取文件路径，比如输入/home/alex/t1.nii.gz返回/home/alex
filename=`basename $1`        # just file name plus extension
## 获取文件名，比如输入/home/alex/t1.nii.gz返回t1.nii.gz
dataname=`${FSLDIR}/bin/remove_ext ${inputfile}`;          # path and file name minus extension
## 去掉文件后缀，比如输入/home/alex/t1.nii.gz返回/home/alex/t1。这个变量似乎并没有作用
filebase=`${FSLDIR}/bin/remove_ext ${filename}`;          # file name minus extension
## 获取文件名，比如输入t1.nii.gz返回t1
workpath=`echo ${fullpath}/${filebase}`                    #make new directory for report
```

```

zmax_all=`$FSLDIR/bin/fslstats $1 -l 0.0001 -R 2>/dev/null | awk '{print $2}'` # Max Z for all
input volume
## 输出最大的值
mkdir ${workpath}
echo 'Working on '$fullpath/${filename}

#Generate report.html
echo "<link REL=stylesheet TYPE=text/css href=/usr/local/fsl/doc/fsl.css>" >>
${workpath}/report.html
## 读取FSL的CSS文件
echo "<Title> Cluster Atlas Query Report beta 1.0</Title>" >> ${workpath}/report.html
## HTML Title元素
echo "<TABLE BORDER=0><TR><TD ALIGN=CENTER WIDTH=100%><TABLE BORDER=0><tr> \
    <td align=center><font size=+3><b>Cluster Atlas Query Report</b></font></tr> \
    <tr><td valign=center height=25 align=center>Worked on ${filename}, threshold at $2, \
cluster size above $3<br> \
    Written by Yazhuo Kong, ykong@fmrib.ox.ac.uk</tr><br></table><TD ALIGN=RIGHT><a \
href=http://www.fmrib.ox.ac.uk/fsl target=_top> \
    <IMG BORDER=0 SRC=file:/usr/local/fsl/doc//images/fsl-logo-big.jpg WIDTH=165></a></TD> \
    </TR></TABLE><hr>" >> ${workpath}/report.html
## 第一个table元素用来表示文字和logo图片，第二个table元素用来表示文字部分的不同内容。似乎少了两个</TD>标签。
# tmpfile - report in MNI coordinates, tmpfile1 in subject
$FSLDIR/bin/cluster -i $1 -t $2 --mm > tmpfile.txt
$FSLDIR/bin/cluster -i $1 -t $2 > tmpfile1.txt
## 使用cluster报告超过阈值的团块和坐标信息（mm坐标和体素坐标，后面mm坐标用于报告脑区，体素坐标用于画图）
/bin/cat tmpfile.txt | awk {'print $2'} > tmp_cluster_size.txt
$FSLDIR/bin/overlay 1 0 $FSLDIR/data/standard/MNI152_T1_2mm_brain -a $1 $2 ${zmax_all}
tmp_overlay
## 提取cluster大小的信息，并将卡过阈值以后的文件和MNI152模板叠加在一起用于后面进行可视化（阈值和最大值作为colorbar的范围）
row=1
cluster_index=0
for voxels in ` /bin/cat tmp_cluster_size.txt ` ; do
    # Ignore the first row of index texts
    if [ ${row} -ge 2 ] ; then
        # Only check the atlas for cluster size over $3
        if [ ${voxels} -ge $3 ] ; then
            z_max=`/bin/cat tmpfile.txt | awk {"if (NR==${row}) print " } | awk {'print $3}'`
            x_index=`/bin/cat tmpfile.txt | awk {"if (NR==${row}) print " } | awk {'print $4}'`
            y_index=`/bin/cat tmpfile.txt | awk {"if (NR==${row}) print " } | awk {'print $5}'`
            z_index=`/bin/cat tmpfile.txt | awk {"if (NR==${row}) print " } | awk {'print $6}'`
            ## 如果团块大小大于设定的值，提取该团块的Peak值以及X/Y/Z坐标
            cluster_index=`echo ${row} -1 | bc`
            echo "<h3>Cluster ${cluster_index}</h3>${voxels} voxels, Max Z is ${z_max}, Peak \
            Voxel MNI coordinates (${x_index} ${y_index} ${z_index}) <br \><br \>" >>
        fi
    fi
    row=$((row+1))
done

```

```

${workpath}/report.html

#echo "Harvard-Oxford Cortical Structural Atlas"
$FSLDIR/bin/atlasquery -a "Harvard-Oxford Cortical Structural Atlas" -c \
${x_index},${y_index},${z_index} >> ${workpath}/report.html
echo "<br \>" >> ${workpath}/report.html
#echo "Harvard-Oxford Subcortical Structural Atlas"
$FSLDIR/bin/atlasquery -a "Harvard-Oxford Subcortical Structural Atlas" -c \
${x_index},${y_index},${z_index} >> ${workpath}/report.html
echo "<br \>" >> ${workpath}/report.html
#echo "MNI Structural Atlas"
$FSLDIR/bin/atlasquery -a "MNI Structural Atlas" -c \
${x_index},${y_index},${z_index} >> ${workpath}/report.html
## 使用atlasquery提取特定坐标的脑区
#generate figure
x_index1=`/bin/cat tmpfile1.txt | awk {"if (NR==${row}) print "}" | awk {'print $4'}`
y_index1=`/bin/cat tmpfile1.txt | awk {"if (NR==${row}) print "}" | awk {'print $5'}`
z_index1=`/bin/cat tmpfile1.txt | awk {"if (NR==${row}) print "}" | awk {'print $6'}`
$FSLDIR/bin/slicer tmp_overlay -s 2 -u -n -x -${x_index1} x.png -y -${y_index1} y.png -z
-${z_index1} z.png
$FSLDIR/bin/pngappend x.png + y.png + z.png ${workpath}/cluster_${cluster_index}.png
/bin/rm x.png y.png z.png
## 根据体素坐标生成X/Y/Z三个截面的图
echo "<br \><br \><IMG BORDER=0 SRC=./cluster_${cluster_index}.png><br \>" >>
${workpath}/report.html

fi
fi
row=`echo ${row} + 1 | bc` ;
done

/bin/rm tmp*

if [ ${cluster_index} -lt 1 ] ; then
echo "No cluster found for threshold $2 and above $3 voxels! Exciting!"
/bin/rm -r ${workpath}
exit 1
else
echo "Finished! Please point your browser at ${workpath}/report.html for the report!"
fi

```

该脚本要求输入位于MNI152 2mm空间，在可视化上不同团块没有分开，不利于可视化。因此，我对该脚本做了一些简单修改：

```

#!/bin/sh

if [ $# -ne 3 ] ; then
    echo "Usage: report_cluster.sh <input volume> <threshold> <cluster size>"
    echo " <1> input volume"
    echo " <2> threshold for input volume"
    echo " <3> cluster size i.e. minimum voxel number in one cluster"
    exit 1
fi

# make temporary folder for intermediate files
inputpath=$(dirname $1)
tmppath=$(echo ${inputpath}/tmp${RANDOM})
mkdir ${tmppath}

# report mm and voxel coordinates, and make cluster mask
cluster -i $1 -t $2 --minextent=$3 --mm -o ${tmppath}/clustermask.nii.gz >
${tmppath}/mmcoord.txt
cluster -i $1 -t $2 --minextent=$3 > ${tmppath}/voxelcoord.txt

NROW=$(cat ${tmppath}/mmcoord.txt | wc -l)
NCLUST=$(( $NROW - 1 ))
if [[ $NCLUST -gt 0 ]]
then
    # resample MNI152 template if input volume is not 2mm
    voxsize=$(fslval $1 pixdim1)
    if [[ $(echo "$voxsize == 2" | bc) -ne 1 ]]
    then
        flirt -in ${FSLDIR}/data/standard/MNI152_T1_2mm_brain.nii.gz -ref $1 -applyxfm -
        usesqform -out ${tmppath}/MNIbrain.nii.gz
    else
        ln -s ${FSLDIR}/data/standard/MNI152_T1_2mm_brain.nii.gz ${tmppath}/MNIbrain.nii.gz
    fi
    # atlas query
    for i in $(seq 1 $NCLUST)
    do
        j=$(( $i + 1 ))
        cluster_index=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $1'})
        cluster_size=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $2'})
        z_max=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $3'})
        x_index=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $4'})
        y_index=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $5'})
        z_index=$(cat ${tmppath}/mmcoord.txt | sed -n "${j}p" | awk {'print $6'})
        echo "<h3>Cluster ${i}</h3>${cluster_size} voxels, Max is ${z_max}, Peak \
        Voxel MNI coordinates (${x_index} ${y_index} ${z_index}) <br \><br \>" >>
        ${inputpath}/report.html
        atlasquery -a "Harvard-Oxford Cortical Structural Atlas" -c \

```

```

    ${x_index},${y_index},${z_index} >> ${inputpath}/report.html
    echo "<br \>" >> ${inputpath}/report.html
    atlasquery -a "Harvard-Oxford Subcortical Structural Atlas" -c \
    ${x_index},${y_index},${z_index} >> ${inputpath}/report.html
    echo "<br \>" >> ${inputpath}/report.html
    atlasquery -a "MNI Structural Atlas" -c \
    ${x_index},${y_index},${z_index} >> ${inputpath}/report.html
    ## generate images
    x_index=$(cat ${tmppath}/voxelcoord.txt | sed -n "${j}p" | awk {'print $4'})
    y_index=$(cat ${tmppath}/voxelcoord.txt | sed -n "${j}p" | awk {'print $5'})
    z_index=$(cat ${tmppath}/voxelcoord.txt | sed -n "${j}p" | awk {'print $6'})
    fslmaths ${tmppath}/clustermask.nii.gz -thr ${cluster_index} -uthr ${cluster_index} -
bin -mul $1 ${tmppath}/cluster${i}.nii.gz
    overlay 1 0 ${tmppath}/MNIbrain.nii.gz -a ${tmppath}/cluster${i}.nii.gz $2 ${z_max}
    ${tmppath}/overlay.nii.gz
    slicer ${tmppath}/overlay.nii.gz -s 2 -u -n -x -${x_index} ${tmppath}/x.png -y
-${y_index} ${tmppath}/y.png -z -${z_index} ${tmppath}/z.png
    pngappend ${tmppath}/x.png + ${tmppath}/y.png + ${tmppath}/z.png
    ${inputpath}/cluster${i}.png
    echo "<br \><br \><IMG BORDER=0 SRC=./cluster${i}.png><br \>" >>
    ${inputpath}/report.html
done
else
    echo "No cluster found for threshold $2 and above $3 voxels! Exciting!"
fi

rm -r ${tmppath}

```