

使用SPM12进行组织分割

Alex / 2023-02-13 / free_learner@163.com / AlexBrain.cn

更新于2023-09-24，主要是文字排版上的更新，内容基本保持不变。

简介使用SPM12进行组织分割的三种方法。

一、背景

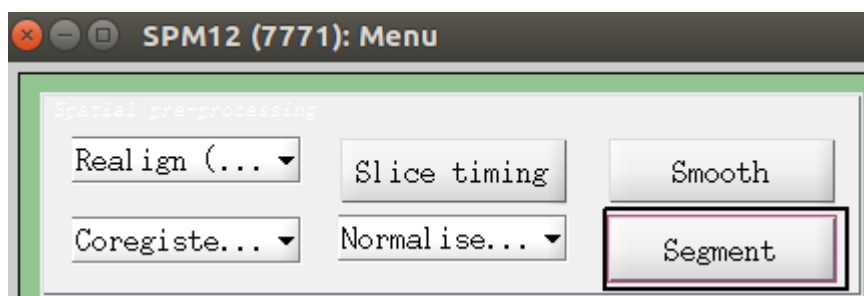
组织分割（tissue segmentation）这里指的是将MRI脑图像分成灰质、白质、脑脊液等不同的组织成分。SPM12的分割算法将偏差场校正（bias field correction）、组织分割和配准（registration）三个过程统一到一个模型中实现，具体请参考：

Ashburner, J., & Friston, K. J. (2005). Unified segmentation. *NeuroImage*, 26(3), 839–851.
<https://doi.org/10.1016/j.neuroimage.2005.02.018>

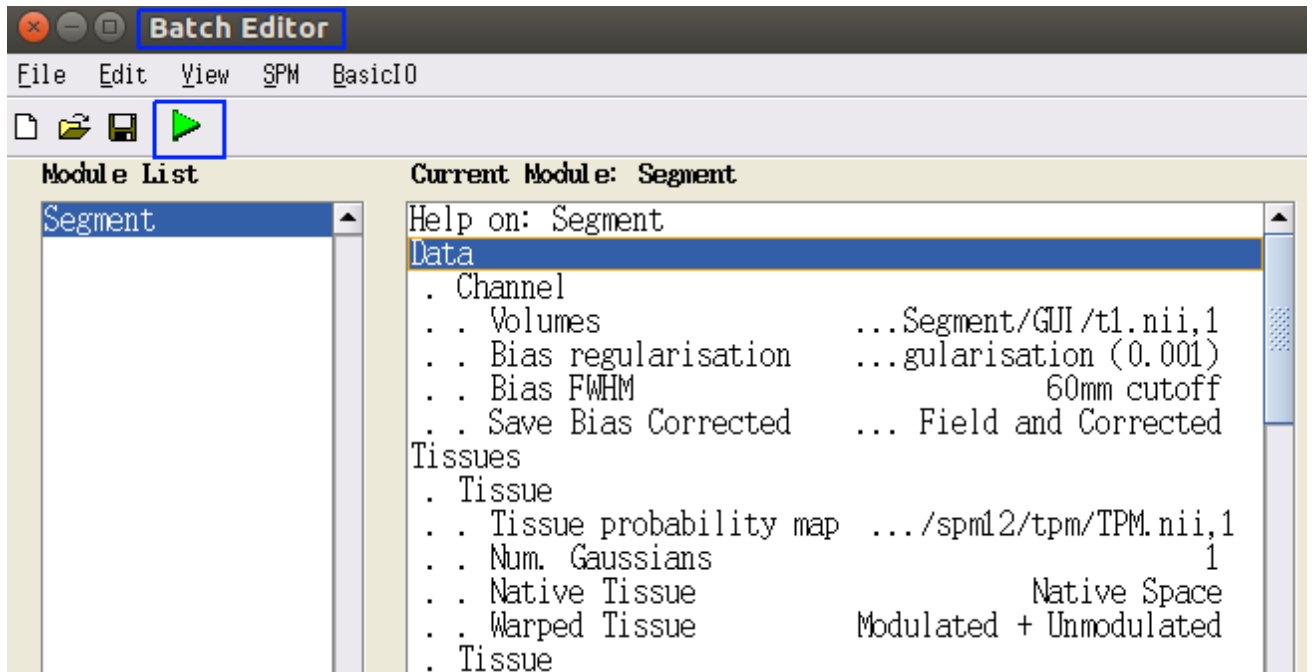
根据我的使用体验，SPM12分割效果非常不错。这里简单介绍一下在SPM12中使用图像界面（GUI）和使用脚本进行组织分割的方法。SPM12的下载、安装和依赖环境请参考：
<https://www.fil.ion.ucl.ac.uk/spm/software/spm12/>，我这里的测试环境是SPM12（7771）、Matlab 2015b、Ubuntu 16.04。

二、使用图形界面

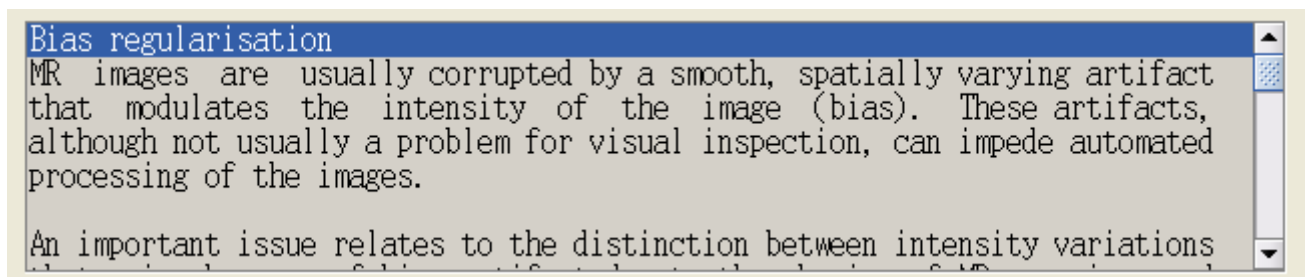
1. 在Matlab命令行输入 `spm_fmri`，在弹出的菜单界面（Menu）中，选择Segment模块，如下图所示：



2. 在弹出的Segment设置界面（Batch Editor）中，选择输入数据（这里假设是T1加权结构像，文件名为 `t1.nii`）和设置参数，如下图所示：



- 完成设置以后（记得保存设置 `File -> Save Batch`，方便以后回溯），选择设置界面上的绿色三角形运行程序，如上图所示。
- 根据具体的参数设置会生成不同类型的文件（输出在输入数据所在的目录下），常用的比如：`BiasField_t1.nii` 表示估计出的偏差场、`mt1.nii` 表示偏差场校正后的图像、`c[1-5]t1.nii` 表示原始空间的5种组织的概率图像、`[m]wc1t1.nii` 表示MNI空间的灰质概率图像，其中 `m` 表示校正了由于配准导致的体积变化、`[i]y_t1.nii` 表示配准产生的（逆）变形场、`t1_seg8.mat` 这个文件可以用于估计组织体积和颅内总体积。
- 关于设置界面中不同选项的含义，在窗口下方会有详细解释（如下图所示），这里不再赘述。



三、使用批处理脚本

当数据比较多的时候，使用图形界面就显得非常繁琐。接下来介绍如何使用脚本来完成上述操作。有两种使用脚本的方式，一种是通过批处理系统（Batch System），一种是使用更底层的函数。我自己对批处理系统的工作原理也不了解。使用批处理脚本的方法很简单，在设置完图形界面以后，选择 `File -> Save Batch and Script`，会生成 `*_job.m` 的文件，我们只需要根据需求修改这个脚本文件即可。作为例子，以下是我修改后的文件：

```

%% initialize SPM12
spm('defaults', 'FMRI');
spm_jobman('initcfg');
%% set matlabbatch
matlabbatch{1}.spm.spatial.preproc.channel.vols =
{'/home/alex/Projects/Testing/SPM/Segment/BATCH/t1.nii,1'};
matlabbatch{1}.spm.spatial.preproc.channel.biasreg = 0.001;
matlabbatch{1}.spm.spatial.preproc.channel.biasfwhm = 60;
matlabbatch{1}.spm.spatial.preproc.channel.write = [1 1];
matlabbatch{1}.spm.spatial.preproc.tissue(1).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,1'};
matlabbatch{1}.spm.spatial.preproc.tissue(1).ngaus = 1;
matlabbatch{1}.spm.spatial.preproc.tissue(1).native = [1 0];
matlabbatch{1}.spm.spatial.preproc.tissue(1).warped = [1 1];
matlabbatch{1}.spm.spatial.preproc.tissue(2).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,2'};
matlabbatch{1}.spm.spatial.preproc.tissue(2).ngaus = 1;
matlabbatch{1}.spm.spatial.preproc.tissue(2).native = [1 0];
matlabbatch{1}.spm.spatial.preproc.tissue(2).warped = [0 0];
matlabbatch{1}.spm.spatial.preproc.tissue(3).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,3'};
matlabbatch{1}.spm.spatial.preproc.tissue(3).ngaus = 2;
matlabbatch{1}.spm.spatial.preproc.tissue(3).native = [1 0];
matlabbatch{1}.spm.spatial.preproc.tissue(3).warped = [0 0];
matlabbatch{1}.spm.spatial.preproc.tissue(4).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,4'};
matlabbatch{1}.spm.spatial.preproc.tissue(4).ngaus = 3;
matlabbatch{1}.spm.spatial.preproc.tissue(4).native = [1 0];
matlabbatch{1}.spm.spatial.preproc.tissue(4).warped = [0 0];
matlabbatch{1}.spm.spatial.preproc.tissue(5).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,5'};
matlabbatch{1}.spm.spatial.preproc.tissue(5).ngaus = 4;
matlabbatch{1}.spm.spatial.preproc.tissue(5).native = [1 0];
matlabbatch{1}.spm.spatial.preproc.tissue(5).warped = [0 0];
matlabbatch{1}.spm.spatial.preproc.tissue(6).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,6'};
matlabbatch{1}.spm.spatial.preproc.tissue(6).ngaus = 2;
matlabbatch{1}.spm.spatial.preproc.tissue(6).native = [0 0];
matlabbatch{1}.spm.spatial.preproc.tissue(6).warped = [0 0];
matlabbatch{1}.spm.spatial.preproc.warp.mrf = 1;
matlabbatch{1}.spm.spatial.preproc.warp.cleanup = 1;
matlabbatch{1}.spm.spatial.preproc.warp.reg = [0 0.001 0.5 0.05 0.2];
matlabbatch{1}.spm.spatial.preproc.warp.affreg = 'mni';
matlabbatch{1}.spm.spatial.preproc.warp.fwhm = 0;
matlabbatch{1}.spm.spatial.preproc.warp.samp = 3;
matlabbatch{1}.spm.spatial.preproc.warp.write = [1 1];
matlabbatch{1}.spm.spatial.preproc.warp.vox = NaN;
matlabbatch{1}.spm.spatial.preproc.warp.bb = [NaN NaN NaN
NaN NaN NaN];

%% run matlabbatch
spm_jobman('run', matlabbatch);

```

四、使用spm_preproc_run函数

我们也可以直接运行 `spm_preproc_run` 函数来进行组织分割。在使用图形界面或批处理脚本时，在Matlab的命令行窗口会输出所调用的函数名字，通过查看函数帮助了解如何使用函数。以下是我自己测试的脚本：

```
%% set parameters
job.channel(1).vols = {'/home/alex/Projects/Testing/SPM/Segment/FUNC/t1.nii,1'};
job.channel(1).biasreg = 0.001;
job.channel(1).biasfwhm = 60;
job.channel(1).write = [1 1];
job.tissue(1).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,1'};
job.tissue(1).ngaus = 1;
job.tissue(1).native = [1 0];
job.tissue(1).warped = [1 1];
job.tissue(2).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,2'};
job.tissue(2).ngaus = 1;
job.tissue(2).native = [1 0];
job.tissue(2).warped = [0 0];
job.tissue(3).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,3'};
job.tissue(3).ngaus = 2;
job.tissue(3).native = [1 0];
job.tissue(3).warped = [0 0];
job.tissue(4).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,4'};
job.tissue(4).ngaus = 3;
job.tissue(4).native = [1 0];
job.tissue(4).warped = [0 0];
job.tissue(5).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,5'};
job.tissue(5).ngaus = 4;
job.tissue(5).native = [1 0];
job.tissue(5).warped = [0 0];
job.tissue(6).tpm = {'/home/alex/matlab/spm12/tpm/TPM.nii,6'};
job.tissue(6).ngaus = 2;
job.tissue(6).native = [0 0];
job.tissue(6).warped = [0 0];
job.warp.mrf = 1;
job.warp.cleanup = 1;
job.warp.affreg = 'mni';
job.warp.reg = [0 0.001 0.5 0.05 0.2];
job.warp.fwhm = 0;
job.warp.samp = 3;
job.warp.write = [1 1];
job.warp.bb = [NaN NaN NaN; NaN NaN NaN];
job.warp.vox = NaN;
```

```
%% run the core function  
spm_preproc_run(job)
```