

使用SimTB生成模拟fMRI数据

Alex / 2018-12-31 / free_learner@163.com / learning-archive.org

更新于2023-07-28，主要是文字排版上的更新，内容基本保持不变。

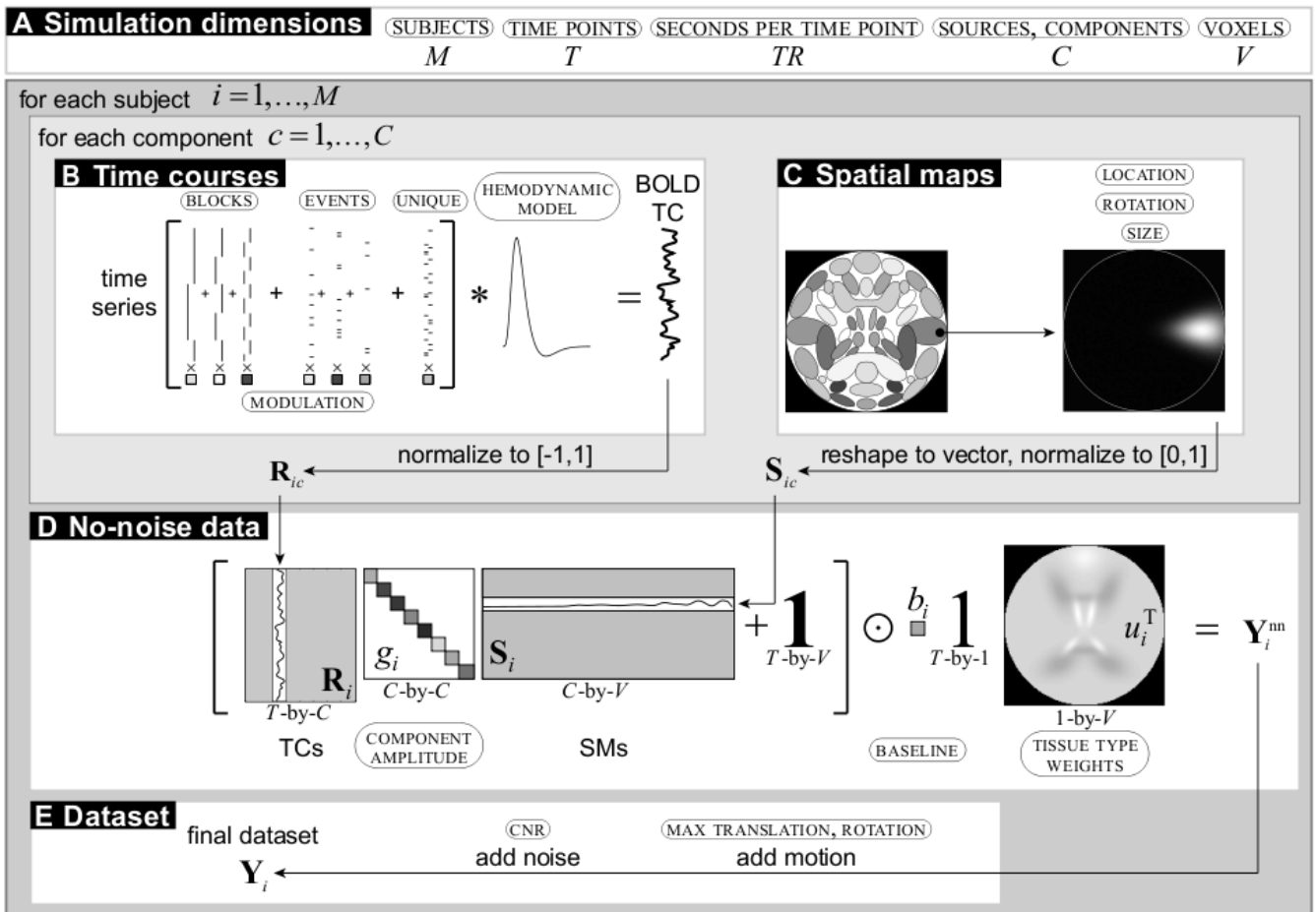
SimTB是一个生成模拟fMRI数据的MATLAB工具包，这里介绍它的原理和基本用法。

一、下载和安装

[下载](#)SimTB压缩包，解压后将其文件夹和子文件夹加入MATLAB搜索路径即可。

二、原理

SimTB生成模拟数据的原理是假设fMRI数据（人类脑活动）可以分解为时间进程和空间成分两部分。下面我试图用一个简单的例子（而不是用数学公式）来说明什么是时间-空间可分性：（我不保证接下来内容的准确性）现实世界纷繁复杂，哲学家们认为这只是表面现象，他们试图找到一些基本元素，这些元素的复杂组合构成了现实事物。比如，“五行学说”认为世界由五种基本元素组成（金、木、水、火、土），这些元素（假设）按不同比例合成，就产生了世界万物。不同元素的比例随着时间而变化，世界万物就随时间而生灭消长。fMRI数据（人类脑活动）同样可以看作是由不同元素构成并且随着时间变化的，这里五行就变成了不同的脑区或网络，而这些不同脑区或网络随时间的变化就产生了随着时间变化的大脑活动。因此，空间成分就是将大脑分成几个基本单元，时间进程就是大脑的基本单元随着时间的变化模式，不同空间成分随时间的变化叠加在一起就构成了我们使用fMRI技术观测到的信号。



上图来自于SimTB官方文档，图C就是30个空间成分，图B就是三种类型的时间进程（随不同类型的实验设计而改变，比如block/event/rest等），图D就是将不同空间成分随时间的变化叠加在一起，就得到了没有噪声的数据，图E加上头动噪声和随机噪声等，就生成了最终的模拟数据。

三、使用脚本生成模拟数据

SimTB可以使用GUI和脚本来生成模拟数据，这里使用脚本的方式。在官方文档里给出了一个比较复杂例子，我这里模拟一个比较简单的情况：3个被试的静息态fMRI数据，其中每个被试包含5个空间成分。在SimTB包中examples文件夹下有个名为 `experiment_params_aod.m` 的文件，该文件包含了所有的参数，只需要根据我们的需求更改即可，假设修改后的参数文件名为 `sim_param.m`。

1. 输出参数

```

out_path = '/home/Alex/SimTB/OUTPUT'; % 输出目录
prefix = 'rest'; % 输出文件前缀
saveNII_FLAG = 1; % 结果保存为NIFTI格式
verbose_display = 0; % 模拟过程中不弹出图片

```

2. 随机数选项

```
seed = 3571; % 固定seed, 使得多次模拟结果完全一样
simtb_rand_seed(seed);
```

3. 维度参数

```
M = 3; % 被试数目
nV = 148; % 空间成分矩阵nV*nV
nT = 150; % 时间点个数
TR = 2; % 重复时间
```

4. 空间成分

```
SM_source_ID = [8 13 15 19 22]; % 不同空间成分对应的标签,
% 可以使用simtb_pickSM选择
nC = length(SM_source_ID); % 空间成分个数
SM_present = ones(M,nC); % 空间成分是否在每个被试中都出现
SM_translate_x = 1*randn(M,nC); % X方向位移, 均值0, 方差1个体素
SM_translate_y = 1*randn(M,nC); % Y方向位移, 均值0, 方差1个体素
SM_theta = 10*randn(M,nC); % 旋转, 均值0, 方差10度 (degree)
SM_spread = 1+0.1*randn(M,nC); % 缩放, 均值1, 方差0.1
D_baseline = 800*ones(1,M); % 基线值
D_TT_FLAG = 0; % 0表示基线值对于不同组织是固定的
```

5. 时间进程

```
TC_source_type = ones(1,nC); % 1表示与标准HRF函数进行卷积
TC_source_params = cell(M,nC); % HRF函数的参数, 不设置表示使用默认参数
TC_unique_FLAG = 1; % 1表示随机产生event, 适用于静息态
TC_unique_prob = 0.2*ones(1,nC); % 每个TR产生随机event的概率为0.2
TC_unique_amp = ones(M,nC); % 随机event的幅值
D_pSC = 3 + 0.25*randn(M,nC); % 信号变化百分比, 均值为3 (3%), 方差0.25
```

6. 噪声

```
D_noise_FLAG = 1; % 1表示添加随机噪声
D_CNR = rand(1,M)*(2-0.65) + 0.65; % 信噪比在0.65-2之间均匀分布
D_motion_FLAG = 0; % 0表示不添加头动噪声
```

7. 执行脚本

```

sP = simtb_create_sP('sim_para'); % 生成参数变量sP
simtb_main(sP); % 生成模拟数据
simtb_createmask(sP, 1); % 生成mask文件

```

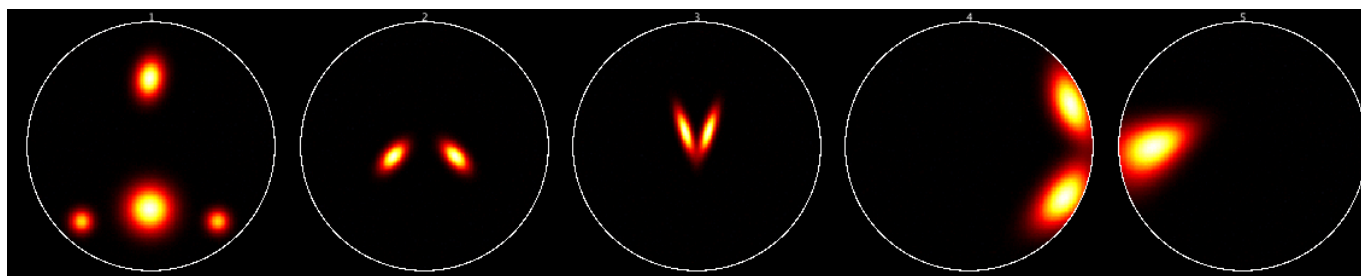
四、可视化

1. 一个被试的所有空间成分

```

SM = simtb_group_getSM(sP, 1); % 提取第一个被试的所有成分
SM = squeeze(SM);
simtb_showSM(SM, 'l', 'l', 'l', 0);

```

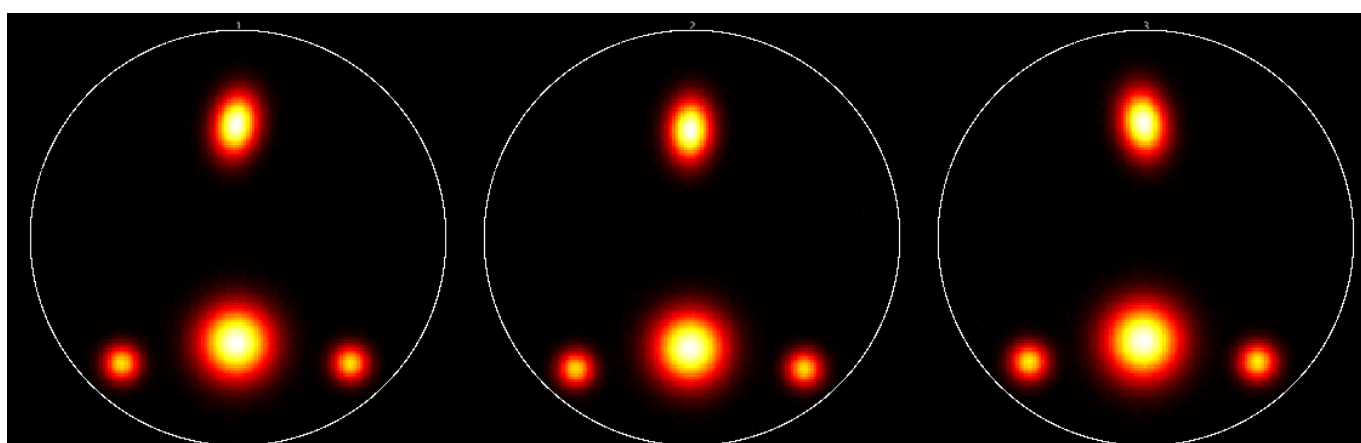


2. 所有被试的第一个空间成分

```

SM = simtb_group_getSM(sP); % 提取所有被试的第一个成分
SM = squeeze(SM(:,1,:));
simtb_showSM(SM, 'l', 'l', 'l', 0);

```

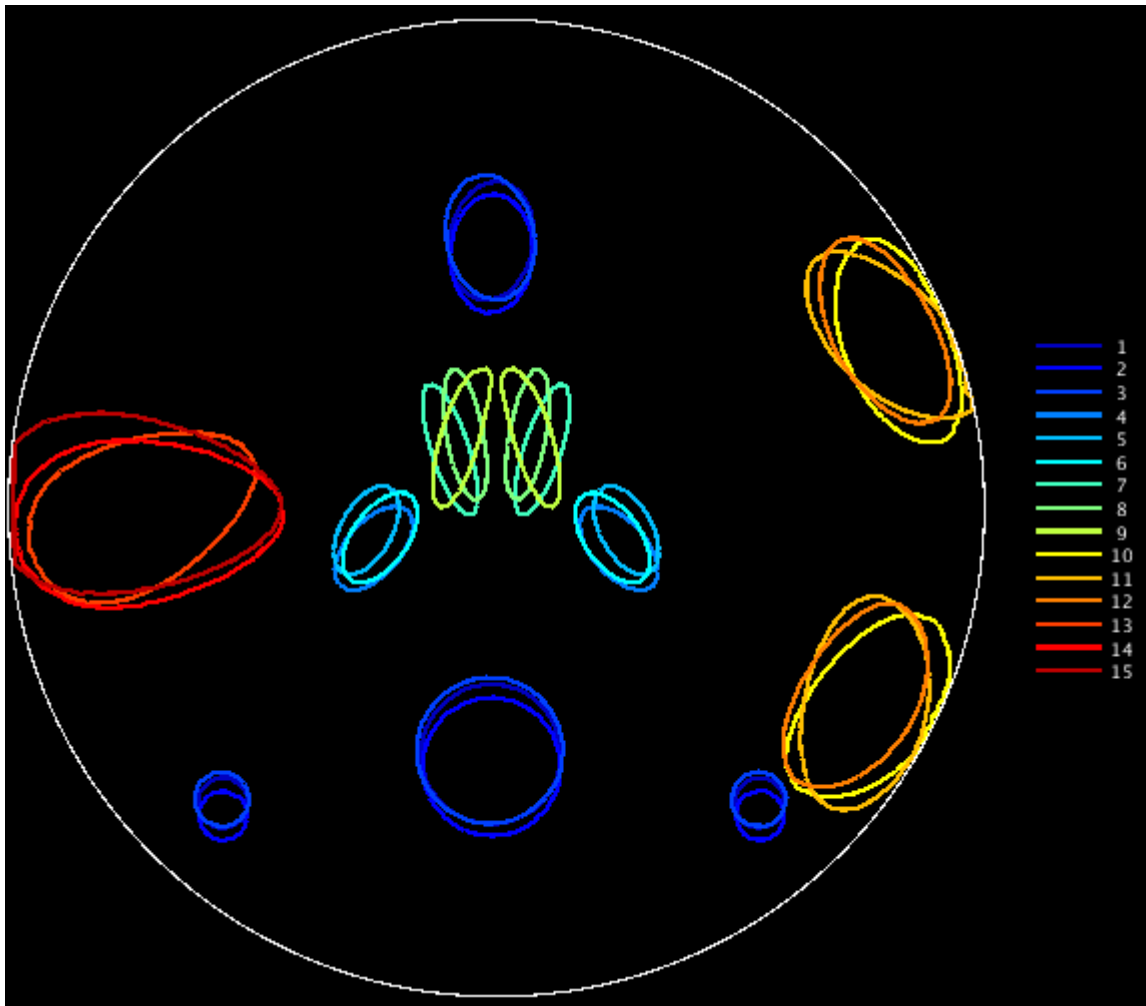


3. 所有被试所有成分的轮廓

```

cmap = jet(15);
SM = simtb_group_getSM(sP);
SM = reshape(SM, size(SM,1) * size(SM,2), size(SM,3));
simtb_showSMContours(SM,0.5, cmap,0);

```



4. 所有被试时间进程的相关系数

```

TC = simtb_group_getTC(sP);
TC = permute(TC, [1 3 2]);
TC = reshape(TC, size(TC,1) * size(TC,2), size(TC,3));
Cor = corr(TC');
imagesc(Cor);
colorbar;

```

