

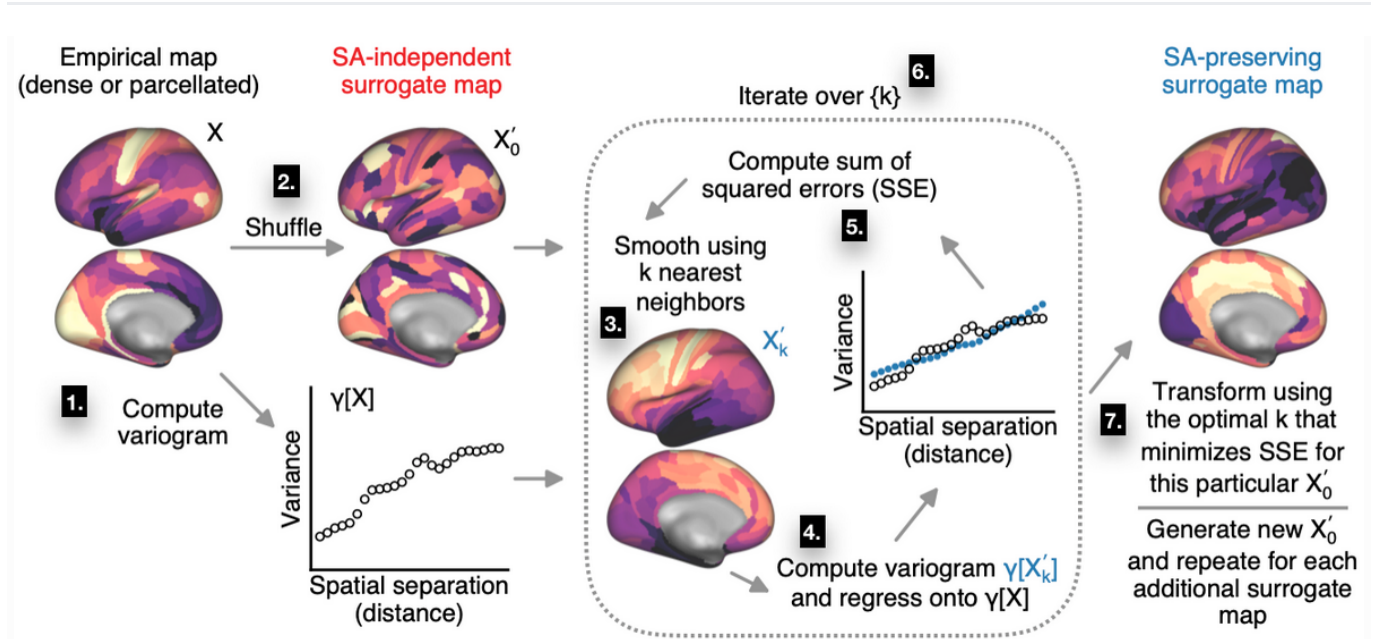
使用BrainSMASH进行空间相关

Alex / 2023-07-02 / free_learner@163.com / AlexBrain.cn

一、背景

在进行不同脑图间的关联分析的时候，需要考虑到脑图本身的空间自相关性（spatial autocorrelation）。这里介绍使用Python环境下的BrainSMASH包进行空间相关的基本方法。所有内容来自于官方[使用文档](#)。

二、原理



如上图所示，brainSMASH首先计算真实图像的空间自相关性（通过variogram来度量），然后置换不同脑区得到代理图像（surrogate map），对代理图像进行一定程度的平滑使得代理图像和真实图像的空间自相关性匹配。通过生成一定数量的代理图像，就可以用来构造统计量的零分布了。更详细的原理请参考：

Burt, J. B., Helmer, M., Shinn, M., Anticevic, A., & Murray, J. D. (2020). Generative modeling of brain maps with spatial autocorrelation. *NeuroImage*, 220, 117038.

三、使用BrainSMASH

1. 安装BrainSMASH

Python版本要求3+，我这里测试的Python版本是3.7.16，BrainSMASH版本是0.11.0。

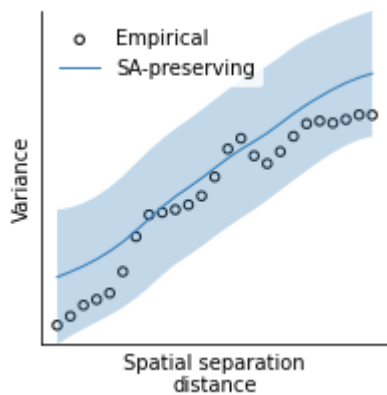
```
pip install brainsmash
```

2. 一个使用样例

```
import numpy as np
from brainsmash.mapgen.base import Base
from brainsmash.mapgen.eval import base_fit
from brainsmash.mapgen.stats import pearsonr
from brainsmash.mapgen.stats import nonparp
from scipy import stats
## load data
myelin_map = np.loadtxt('LeftParcelMyelin.txt')
dist_mat = np.loadtxt('LeftParcelGeodesicDistmat.txt')
thickness_map = np.loadtxt('LeftParcelThickness.txt')
## check the fit and tune the parameters
base_fit(myelin_map, dist_mat, nsurr=1000, kernel='exp', pv=25, nh=25, resample=False,
seed=100, n_jobs=4)
## generate surrogate data
gensurrmap = Base(x=myelin_map, D=dist_mat, kernel='exp', pv=25, nh=25, resample=False,
seed=100, n_jobs=4)
surrogate_maps = gensurrmap(n=1000)
## correlation between surrogate myelin data and thickness data
surrogate_brainmap_corrs = pearsonr(thickness_map, surrogate_maps).flatten()
## correlation between real myelin and thickness data
real_corr = stats.pearsonr(myelin_map, thickness_map)[0]
## calculate the p-value
all_corrs = np.concatenate([np.array([real_corr]), surrogate_brainmap_corrs])
pval = nonparp(test_stat, all_corrs)
```

我在这个测试中，使用了BrainSMASH提供的[样例数据](#)，分别是左侧皮层180个脑区的髓鞘（LeftParcelMyelin.txt）和皮层厚度（LeftParcelThickness.txt），以及180个脑区间的距离矩阵（LeftParcelGeodesicDistmat.txt）。

在正式生成代理数据前，我先用 `base_fit` 来调试参数，最后会得到代理数据和真实数据自相关的匹配程度，如下图所示：



四、其他

这里测试的是脑区（parcel）为单位的数据，BrainSMASH针对体素或者顶点数据（dense maps）有相应的优化，我这里并没有测试。