neuromaps包的基本用法

Alex / 2026-11-23 / free_learner@163.com / AlexBrain.cn

本文介绍Python环境下的neuromaps包的基本用法，包括获取脑注释图、不同坐标空间的转换、根据空间零模型计算p值等。

一、安装neuromaps

1. 安装neuromaps

```
pip install neuromaps
pip install brainsmash
pip install brainspace
```

我当前测试的版本是0.0.7。

2. 安装Connectome Workbench

在以前的博客中有介绍，这里不再赘述。可以运行下面的命令查看是否安装成功：

```
wb_command -version
```

3. 设置数据目录

设置neuromaps下载的模板（atlas）和注释（annotations）文件存放的位置，默认是在`$HOME/neuromaps-data`。可以通过设置环境变量`$NEUROMAPS_DATA`来指定其他位置。注意：annotation在FreeSurfer里指的是分区模板，但是在neuromaps里指的是反映大脑特征的空间图谱。

二、坐标系统

neuromaps支持MNI152、fsaverage、fsLR、CIVET、BigBrain五种坐标系统，并提供了不同的分辨率。具体如下：

空间	类型	支持的分辨率
MNI152	体积（NIfTI）	1mm、2mm、3mm
fsLR	表面（GIFTI）	4k、8k、32k、164k（每半球顶点数）
fsaverage	表面（GIFTI）	1k、3k、10k、41k、164k
CIVET	表面（GIFTI）	41k
BigBrain	表面（GIFTI）	164k

其中，MNI152特指由FSL提供的MNI ICBM 152 non-linear 6th generation asymmetric template版本。这意味着，不同坐标系统或者同一坐标系统不同分辨率的数据可以进行相互转换。同时，在获取脑注释图的时候，也要明确数据所在空间以确定是否需要进一步转换。注意：当前版本的neuromaps只能实现从MNI152转换到皮层表面空间，无法实现从皮层表面空间转换到MNI152。

三、获取脑注释图

通过下面的代码查看和获取感兴趣的脑注释图：

```
from neuromaps.datasets import available_annotations, fetch_annotation
for annotation in available_annotations():
    print(annotation)
abagen = fetch_annotation(source='abagen')
```

其中，`available_annotations`返回所有的注释图，每个注释图是包含4个元素的元组，比如：`('abagen', 'genepc1', 'fsaverage', '10k')`，第一个元素表示注释图的来源（source）；第二个元素表示注释图的一个简单说明，比如`genepc1`表示基因表达的第一个主成分，更详细的说明需要去查看每个注释图对应的原始文献；第三个元素表示注释图所在的坐标系统；第四个元素表示注释图的分辨率。`fetch_annotation`下载指定的注释图，返回注释图的路径；如果已经下载过，就会自动跳过下载。

四、转换坐标空间和重采样分辨率

通过下面的代码可以实现不同空间的转换以及改变分辨率：

```
from neuromaps import transforms
## Resampling data from MNI152 2mm to MNI152 3mm
neurosynth_3mm = transforms.mni152_to_mni152(neurosynth, '3mm')
## Transform data from MNI152 to fsLR
neurosynth_fslr = transforms.mni152_to_fslr(neurosynth, '32k')
## Transform data from fsaverage to fsLR
abagen_fslr = transforms.fsaverage_to_fslr(abagen, '32k')
```

默认情况下，这些函数采用线性插值，可以通过`method='nearest'`参数使用最紧邻插值。除了分别转换单个注释图，还可以通过下面的代码同时实现两个注释图的转换：

```
from neuromaps.resampling import resample_images
## Transform neurosynth data from MNI152 to fsaverage and keep abagen unchanged
neurosynth_res, abagen_res = resample_images(src=neurosynth, trg=abagen, src_space='MNI152',
trg_space='fsaverage', method='linear', resampling='downsample_only')
## Transform both neurosynth and abagen data to fsLR space
neurosynth_res, abagen_res = resample_images(src=neurosynth, trg=abagen, src_space='MNI152',
trg_space='fsaverage', method='linear', resampling='transform_to_alt', alt_spec=('fslr',
'32k'))
```

其中，`resampling=downsample_only`表示总是将高分辨率重采样到低分辨率，但是如果一个空间是MNI152，另一个空间是皮层表面，那么总是将MNI152转换到皮层表面。`resampling='transform_to_alt'`表示将两个注释图转换到第三个空间作为公共的空间，同时需要通过`alt_spec=('fslr', '32k')`参数来指定第三个空间。

五、计算空间相关和p值

通过下面的代码计算两个注释图的空间相关以及使用空间零模型计算p值：

```
from neuromaps import nulls, stats
rotated = nulls.alexander_bloch(neurosynth_res, atlas='fslr', density='32k', n_perm=1000,
seed=1234)
corr, pval = stats.compare_images(neurosynth_res, abagen_res, nulls=rotated)
```

其中，`nulls.alexander_bloch`表示使用`alexander_bloch`算法（Alexander-Bloch et al., 2018）生成零模型，关于neuromaps中包含的各种不同的空间零模型方法，请参考Markello & Misic (2021)，这里不再赘述。`stats.compare_images`默认计算皮尔逊相关系数；注意：`nulls=rotated`必须对应于`stats.compare_images`的第一个参数所指定的数据。

参考文献：

- Alexander-Bloch, A. F., Shou, H., Liu, S., Satterthwaite, T. D., Glahn, D. C., Shinohara, R. T., ... & Raznahan, A. (2018). On testing for spatial correspondence between maps of human brain structure and function. Neuroimage, 178, 540-551.
- Markello, R. D., & Misic, B. (2021). Comparing spatial null models for brain maps. NeuroImage, 236, 118052.

六、处理分区数据

下面这个例子给出了处理分区数据的方法：

```
from neuromaps.images import annot_to_gifti
from neuromaps.parcellate import Parcellater
## Convert FreeSurfer parcellation to GIFTI format
lh_annot = "/usr/local/freesurfer/7.4.1/subjects/fsaverage5/label/lh.aparc.annot"
rh_annot = "/usr/local/freesurfer/7.4.1/subjects/fsaverage5/label/rh.aparc.annot"
lh_parc, rh_parc = annot_to_gifti((lh_annot, rh_annot))
DK = (lh_parc, rh_parc)
## Resample data to a common space
abagen = fetch_annotation(source='abagen')
neurosynth = fetch_annotation(source='neurosynth')
neurosynth_res, abagen_res = resample_images(src=neurosynth, trg=abagen, src_space='MNI152',
trg_space='fsaverage', method='linear', resampling='transform_to_alt', alt_spec=('fsaverage',
'10k'))
## Extract mean data based on a parcellation
parc = Parcellater(DK, "fsaverage")
parc.fit()
neurosynth_parc = parc.transform(neurosynth_res, "fsaverage")
abagen_parc = parc.transform(abagen_res, "fsaverage")
## Generate spatial nulls and calculate correlation and p-value
rotated_neurosynth = nulls.vasa(neurosynth_parc, atlas="fsaverage", density="10k",
parcellation=DK, n_perm=1000, seed=1234)
corr, pval = stats.compare_images(neurosynth_parc, abagen_parc, nulls=rotated_neurosynth)
```

其中，`annot_to_gifti`将FreeSurfer的分区文件转换成neuromaps要求的GIFTI格式；`Parcellater`提供了提取分区数据的一系列功能；如果分区是CIFTI格式，可以使用`dlabel_to_gifti`。

七、可视化

neuromaps也提供了简单的可视化功能，可以用于快速查看数据。基本用法如下：

```
from neuromaps.plotting import plot_surf_template
## Plot vertex-level data
fig = plot_surf_template(abagen_res, template='fsaverage', density='10k', surf='inflated',
cmap='RdBu_r', colorbar=True)
fig.savefig('abagen_genepc1.png', dpi=300, bbox_inches='tight')
## Plot parcellated data
vertex_dat = parc.inverse_transform(abagen_parc)
fig = plot_surf_template(vertex_dat, template='fsaverage', density='10k', surf='inflated',
cmap='RdBu_r', colorbar=True)
fig.savefig('abagen_genepc1_parc.png', dpi=300, bbox_inches='tight')
```

参考来源：https://netneurolab.github.io/neuromaps/index.html