

cortical_surface_plots

June 15, 2023

```
[1]: %load_ext autoreload
      %autoreload 2
```

```
[2]: import warnings
      warnings.filterwarnings('ignore')
```

```
[3]: import numpy as np
      import nibabel as nib
      import pandas as pd

      from surfplot import Plot
      import matplotlib as mpl
      import matplotlib.pyplot as plt
      import seaborn as sns

      from brainspace.utils.parcellation import reduce_by_labels

      import utilities

      import pingouin as pg
      from statannotations.Annotator import Annotator
```

```
[4]: def reduce_data(data, subfields):
      data_reshape = data.reshape(
          data.shape[0], data.shape[1]*data.shape[2]
      )

      subfields_mean = reduce_by_labels(data_reshape, subfields, axis=1)
      subfields_mean_reshape = subfields_mean.reshape(
          subfields_mean.shape[0], data.shape[1], data.shape[2]
      )

      return data_reshape, subfields_mean_reshape
```

```
[58]: save_fig = True
```

Subjects

```
[6]: df = pd.read_csv('../config/participants.txt', dtype=str)
subjects = df.participant_id.to_list()
subjects = [ s for s in subjects if s != '09' ]
```

Cortical

```
[19]: # Discrete colormap
cmap      = plt.cm.summer
cmaplist = [cmap(i) for i in range(cmap.N)]
cmap      = mpl.colors.LinearSegmentedColormap.from_list(
    'Test', cmaplist, len(subjects)
)
```

```
[20]: # Paths to surfaces
ctx_lh = '../results/hcp_mmp/sub-01/lh.inflated.32k_fs_LR.surf.gii'.
    ↪format(subject)
ctx_rh = '../results/hcp_mmp/sub-01/rh.inflated.32k_fs_LR.surf.gii'.
    ↪format(subject)
```

```
[21]: # Path to atlases
ctx_hcp_lh = '../resources/standard_mesh_atlases/lh.hcp-mmp.32k_fs_LR.label.gii'
ctx_hcp_lh = nib.load(ctx_hcp_lh)
ctx_hcp_lh_dict = ctx_hcp_lh.labeltable.get_labels_as_dict()

ctx_hcp_rh = '../resources/standard_mesh_atlases/rh.hcp-mmp.32k_fs_LR.label.gii'
ctx_hcp_rh = nib.load(ctx_hcp_rh)
ctx_hcp_rh_dict = ctx_hcp_rh.labeltable.get_labels_as_dict()
```

```
[22]: v1_mask_lh = np.isin(ctx_hcp_lh.darrays[0].data, [1])
v1_mask_rh = np.isin(ctx_hcp_rh.darrays[0].data, [1])

v2_mask_lh = np.isin(ctx_hcp_lh.darrays[0].data, [4])
v2_mask_rh = np.isin(ctx_hcp_rh.darrays[0].data, [4])
```

```
[23]: # Cortical metric
ctx_metric = 'CBF'
ctx_label = 'Perfusion (ml/100 g/min)'
ctx_crange = (20,60)
ctx_cmap = 'hot'
```

```
[24]: # Load input, per subject
ctx_data = np.zeros((
    len(nib.load(ctx_lh).get_arrays_from_intent('NIFTI_INTENT_POINTSET')[0].
    ↪data),
    len(subjects),
    len(['lh','rh'])
))
```

```

for s, subject in enumerate(subjects):
    for h, hemi in enumerate(['lh', 'rh']):
        ctx_data[:, s, h] = nib.load(
            '../results/surface_maps/sub-{0}/sub-{0}_hemi-{1}_{2}.32k_fs_LR.
↪shape.gii'.format(
                subject, hemi, ctx_metric
            )
        ).darrays[0].data

# Replace zeros with NaN to ignore
ctx_data[ctx_data==0] = np.nan

```

```

[25]: # Coverage
coverage = np.nan_to_num(ctx_data, nan=0)
coverage[coverage!=0] = 1

ctx_lh_coverage = np.nansum(coverage[:, :, 0], axis=1)
ctx_rh_coverage = np.nansum(coverage[:, :, 1], axis=1)

```

```

[26]: # Plot
views      = ['posterior', 'medial']
elevations = [-50, -30, -30, -50]
zooms      = [1.5, 1.4, 1.4, 1.5]

p = Plot(
    ctx_lh, ctx_rh, layout='row', views=views, zoom=zooms, elevation=elevations,
    size=(1400, 300), mirror_views=True
)

lh_mask = np.where(ctx_lh_coverage==10, 1, 0)
rh_mask = np.where(ctx_rh_coverage==10, 1, 0)

p.add_layer({
    'left': ctx_lh_coverage,
    'right': ctx_rh_coverage,
}, color_range=(1, 10), cmap=cmap)
p.add_layer({
    'left': lh_mask,
    'right': rh_mask
},
as_outline=True, cmap='gray_r',
cbar=False
)

cbar_ctx_kws = dict(
    outer_labels_only=True, pad=.02, n_ticks=2, decimals=0,

```

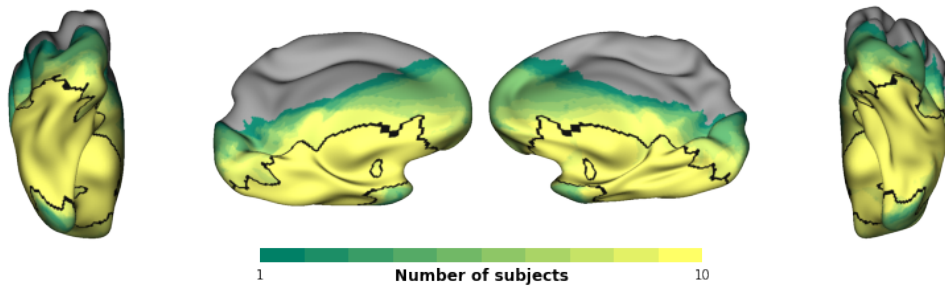
```

        shrink=.4, fraction=.1, aspect=30, draw_border=False
    )

    fig = p.build(cbar_kws=cbar_ctx_kws)
    fig.axes[1].set_xlabel(
        'Number of subjects', labelpad=-11, fontsize=12, fontweight='bold'
    )

    if save_fig:
        filename = '../visualization/neocortical/sub-group_asl_coverage.png'
        fig.savefig(filename, dpi=600, bbox_inches='tight', transparent=True)
    fig.show()

```



```

[27]: # Average across subjects (ignoring NaN vertices)
ctx_lh_cbf = np.nanmean(ctx_data[:, :, 0], axis=1)
ctx_rh_cbf = np.nanmean(ctx_data[:, :, 1], axis=1)

```

```

[28]: ctx_lh_cbf_masked = ctx_lh_cbf.copy()
      ctx_lh_cbf_masked[lh_mask==0] = np.nan

      ctx_rh_cbf_masked = ctx_rh_cbf.copy()
      ctx_rh_cbf_masked[rh_mask==0] = np.nan

```

```

[29]: # Plot
views      = ['posterior', 'medial']
elevations = [-50, -30, -30, -50]
zooms      = [1.5, 1.4, 1.4, 1.5]

p = Plot(
    ctx_lh, ctx_rh, layout='row', views=views, zoom=zooms, elevation=elevations,
    size=(1400, 300), mirror_views=True
)

p.add_layer({'left': ctx_lh_cbf, 'right': ctx_rh_cbf},
            color_range=ctx_crange, cmap=ctx_cmap)
p.add_layer({'left': ctx_lh_cbf/ctx_lh_cbf, 'right': ctx_rh_cbf/ctx_rh_cbf},

```

```

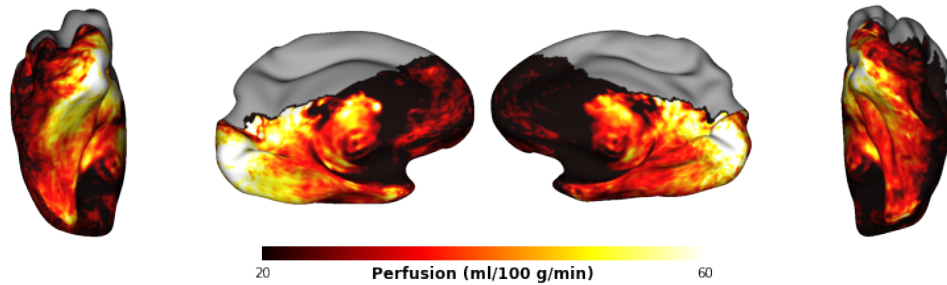
        as_outline=True, cmap=ctx_cmap, cbar=False)

cbar_ctx_kws = dict(
    outer_labels_only=True, pad=.02, n_ticks=2, decimals=0,
    shrink=.4, fraction=.1, aspect=30, draw_border=False
)

fig = p.build(cbar_kws=cbar_ctx_kws)
fig.axes[1].set_xlabel(
    f'{ctx_label}', labelpad=-11, fontsize=12, fontweight='bold'
)

if save_fig:
    filename = f'../visualization/neocortical/sub-group_{ctx_metric}.png'
    fig.savefig(filename, dpi=600, bbox_inches='tight', transparent=True)
fig.show()

```



```

[31]: lhrh_mask = np.logical_and(lh_mask, rh_mask)
      hpc_data = np.nanmean(subj_data[:, :, :, 0], axis=1)

```

```

[60]: # Compare cortical and hippocampal CBF distributions
      fig, ax = plt.subplots(1,1, figsize=(6.5,3))

      # Cortical distribution
      sns.kdeplot(
          x=np.hstack((
              ctx_lh_cbf,
              ctx_rh_cbf
          )),
          color=sns.color_palette()[0],
          ax=ax
      )

      # Hippocampal distribution
      for s in range(subj_data.shape[1]):
          sns.kdeplot(

```

```

        x=np.hstack((
            ctx_data[:,s,0],
            ctx_data[:,s,1]
        )),
        alpha=.2,
        color=sns.color_palette()[0],
        ax=ax
    )

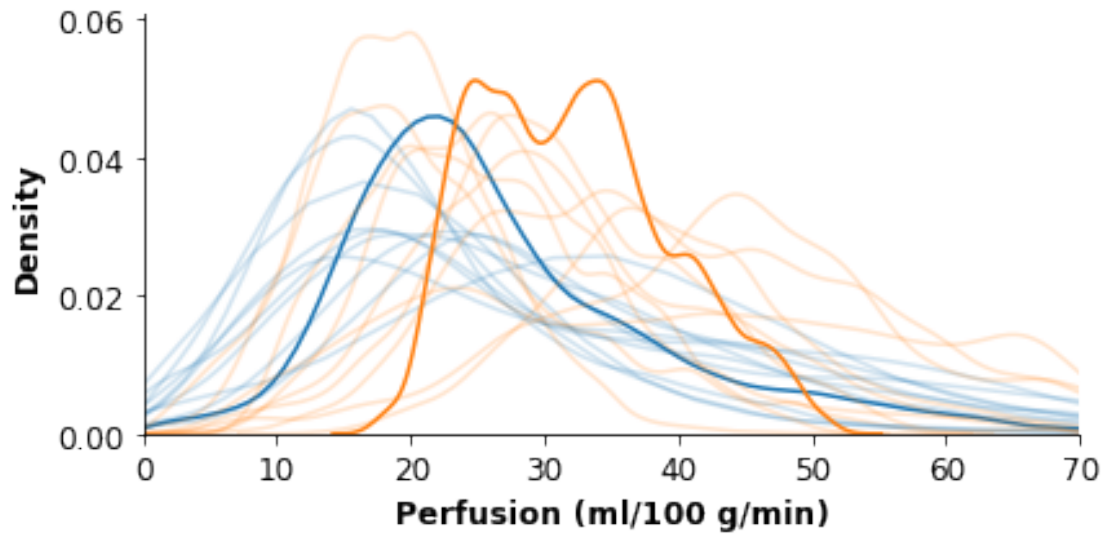
    sns.kdeplot(
        x=np.nanmean(subj_data[:,s,:,0], axis=1),
        alpha=.2,
        color=sns.color_palette()[1],
        ax=ax
    )

sns.kdeplot(
    x=np.nanmean(hpc_data, axis=1),
    color=sns.color_palette()[1],
    ax=ax
)

ax.set_xlim(0,70)
ax.set_xlabel('Perfusion (ml/100 g/min)', weight='bold', fontsize=12)
ax.set_ylabel('Density', weight='bold', fontsize=12)
ax.tick_params(axis='both', labelsize=12)
sns.despine()

if save_fig:
    filename = '../visualization/neocortical/sub-group_cbf_distributions.png'
    fig.savefig(filename, dpi=600, bbox_inches='tight', transparent=True)
plt.show()

```



```
[42]: # V1 stats
v1_data = np.nanmean(
    np.vstack((
        ctx_data[v1_mask_lh,:,0],
        ctx_data[v1_mask_rh,:,1]
    )), axis=1
)

v1_cbf_mean = np.nanmean(
    v1_data
)

v1_cbf_sd = np.nanstd(
    v1_data
)
```

```
[43]: # V2 stats
v2_data = np.nanmean(
    np.vstack((
        ctx_data[v2_mask_lh,:,0],
        ctx_data[v2_mask_rh,:,1]
    )), axis=1
)

v2_cbf_mean = np.nanmean(
    v2_data
)
```

```
v2_cbf_sd = np.nanstd(  
    v2_data  
)
```

```
[44]: print(f'V1: {v1_cbf_mean} +/- {v1_cbf_sd}, V2: {v2_cbf_mean} +/- {v2_cbf_sd}')
```

```
V1: 58.24001193805665 +/- 15.676639138705113, V2: 44.415350621304 +/-  
10.906859966267273
```