Cortical Layer Sampling and Mapping

using BrainVoyager

by Jorie van Haren

March 8, 2024

Contents

0.1 Short word

Below you will find instructions for my pipeline to - using our fine-tuned segmentation - do cortical thickness analyses. Then, using this thickness analysis in combination with a just-created mid-grey matter volume, generate surfaces for the desired number of cortical depths. These surfaces can consequently be used to sample volume VMP files throughout these depths. The current procedure relies on a combination of BrainVoyager 22.2 (Version 22.2.1.4950, 64-bit), BrainVoyager's in house DNN 'Tiramisu' Segmentation (Python evnironment needed) and custom Python BrainVoyager Tools.

- BrainVoyager download page:
- BrainVoyager Python Plugins:

0.2 Splitting Segmentation Files per Hemisphere

The first step we have to undertake is to split our segmentation file into separate files per hemisphere. There are multiple methods to achieve this, but if you are working within the VOI format (like me - taking the DNN segmentation as a starting point), we can simply use the DNN post-processing step to split hemispheres. Subsequently, we can use a custom BrainVoyager Python function to create white-matter grey-matter (WMGM) volumes per hemisphere. Note that this function substitutes the traditional 'Create standard segmentation VMR file' function present within the DNN segmentation postprocessing.

1. Open an ISO_0.4.vmr file of your choice: e.g. *'UNI_reframed_ISO-0.4.vmr'*.
2. Add a region of interest to this file (ctrl+r).
    - Load *'UNI_reframed_ISO-0.4_tissue-labels_CL_RC_LH-RH.voi'* if available.
    - If not run: Volumes - DNN Segmentation Postprocessing - **Disconnect** to Separate left and right hemispheres.
3. Go to Python - Development - **Labelmap_to_WMGM_LHRH.py**.

- Directory, VOI Document and VMR File for Header should be loaded automatically from opened files.
- Running **Convert VOI LH+RH Labels to VMR** should result in a *'WMGM_LH.vmr'* and a *'WMGM_LH.vmr'* file within the Directory Path location.

0.3 Running Cortical Thickness Measurement

Next, we want to estimate cortical thickness based on our WMGM files. Additionally, we want to create a mid-grey matter volume. The following steps we'll have to repeat for both the left and the right hemisphere files (*'WMGM_LH.vmr'* and *'WMGM_RH.vmr'*).

1. Press: Volumes - Cortical Thickness measure.
   - Measurement: Press **GO**.
   - Mid-GM Volume: Press **Create Volume**.
   - Mid-GM Volume: Press **Create VOI**.
2. Lastly, from this Mid-GM file we want to create a suitable mesh.
   - Meshes - Create Mesh - **Reconstruct**.
   - Meshes - Advanced Mesh Smoothing - **Go**.
   - Meshes - Mesh Simplification: Set 'Number of vertices' to 300.000 (check below if filename states 300k as suffix) - **GO**.

0.4 Cortical Depth Sampling

We want to sample our surfaces at different cortical depths using our just-created mid-grey matter surface and our cortical thickness measurement. To achieve this we can use cortical depth sampling. Since our cortical-depth measures are all based on anatomy, the preferred approach here is to sample the depth surfaces only once, and to map our volume maps of interest (our results) to the sampled surfaces later by looping over these depth surfaces.

1. Open the mesh for one hemisphere (later repeat for the other): e.g. *'WMGM_RH_Mid-gm_RECO_D300k.srf'*.
2. Meshes - **Cortical Depth Sampling**.
   - Thickness VMP file: *'WMGM_RH_Thickness.vmp'*.
   - Data VMP file: We'll leave empty, as we don't want to sample our maps just yet. Instead just creating the different surfaces per cortical depth.
   - No. of depths meshes: 11 (or number of depths of choosing).
   - Tick: **Save meshes** checkbox.
   - Tick: **Smooth meshes** checkbox.
   - Leave unticked: 'Create depth surface maps' checkbox.
   - Press **GO**.

0.5 Mapping VMPs to Cortical Depths

After analyses within the volume we can map our volume maps to each of our previously created surfaces. For this step we will utilize a BrainVoyager Python function called **VMP_Cortical_depth.py**: which takes in a VMP file of interest and simply loops over hemispheres and cortical depth maps, and maps the volume map within this depth.

1. Open an ISO_0.4.vm file of your choice, e.g. *'UNI_reframed_ISO-0.4.vmr'*.
2. Link a volume map of interest (VMP) to the opened file (ctrl+m).
3. Python - Python Development - **VMP_Cortical_depth.py**.
    - Directory Path: parent directory (e.g. *'/MRIData/PreProc/S01_SES1/'*).
    - VMR Prefix: automatically loaded - file minus hemisphere suffix (*_LH.vmr* or *_RH.vmr*). Note that this VMR is probably redundant, and only used for the 'linking' of meshes.
    - SRF Prefix: select the desired mesh files to loop over - if left empty will search the current directory for *'_D-'*. Here we will use *'srfs/'* to instead use the sub-directory */srfs/* for the depth files (and selected hemispheres).
    - VMP filename: The desired VMP volume map file to sample throughout the cortical depths. Using *'Betas/PRF.vmp'* will search the sub-directory */Betas/* for the *'PRF.vmp'* volume map.
    - If both hemispheres are selected, the script will loop and sample for both.
    - Interpolation Method is set to trilinear by default (1) but can be set to nearest neighbour by changing this value to 0.
    - Additionally one can choose to sample only nonzero values instead.

0.6 Displaying and Post-processing VMPs and SMPs

Lastly, we can easily use the resulting files for post-analyses in either the volume or on each cortical depth surface. Since we have previously created the VMPs and SMPs for the two hemispheres and at different cortical depths: we can use a simple looping approach to parse and load our data. Below a dummy example of how to load VMP data split by hemispheres and depths.

```
# load some sort of mask
nii = nb.load("segmasks/pp1/Segmentation-label.nii.gz")
nii_data = np.nan_to_num(nii.get_fdata(), nan=0.)
frontal = nii_data == 1

# define our hemispheres and depths (only 3 in this case)
hemispheres = ['LH', 'RH']
depths = [1, 2, 3]

# loop over hemispheres
for i in range(len(hemispheres)):
    print(i, hemispheres[i])

    # loop over depths
    for d in range(len(depths)):
        print(d, depths[d])
```

```
    # load vmp of current hemisphere and depth
    head, img = bvbabel.vmp.read_vmp(join(mridat_dir, pp_dir(1, 1),
vmpfn(hemispheres[i], depths[d])))

    # get index of condition of interest (based on vmp or smp labels
)
    cond_A = cond_names.index('base_U_adaptation (cv mean)')
    cond_B = cond_names.index('prediction (cv mean)')
    cond_AuB = cond_names.index('base_U_adaptation_U_prediction (cv
mean)')

    # sellect only image of that condition
    A = img[:,:,:,cond_A][frontal]
    B = img[:,:,:,cond_B][frontal]
    AuB = img[:,:,:,cond_AuB][frontal]
```