Instructions on data-driven mask usage.

Required code dependencies – ensure all these are added to your matlab path.

Matlab code folder from FreeSurfer, using git on unix terminal, run this command:

```
cd ~/Desktop/dataDriven_largeVessel_segmentation
git clone https://github.com/freesurfer/freesurfer.git
```

The CMRR multi-band pulse sequence physiology file reading:

```
cd ~/Desktop/dataDriven_largeVessel_segmentation
git clone https://github.com/CMRR-C2P/MB.git
```

Add these two folders: 1) freesurfer and 2) MB to your matlab path.

Required MNI 152 1mm files:

1. UBA atlas (provided in this package, named: UBA_exclude_VA_PCOM_mni.nii.gz)
    a. Please note that this atlas was modified from the UBA24 atlas generated by Dunås et al. Please properly cite their work:

Dunås, T., Wåhlin, A., Ambarki, K. et al. A Stereotactic Probabilistic Atlas for the Major Cerebral Arteries. Neuroinform 15, 101–110 (2017). https://doi.org/10.1007/s12021-016-9320-y

The original atlas can be downloaded from: https://www.nitrc.org/projects/brainarteries

2. SSS MNI segmentation (provided in this package, named: MNI152_T1_1mm_SSS.nii.gz).

Required local scan files:

1. fMRI scan
2. fMRI scan finger plethysmography data file (in CMRR physiology format).
    - If you can't read the physiology file with the CMRR read function, you must figure out how to incorporate your physiology finger ppg reading into the code. This version is only compatible with physiology files that can be read with CMRR.
    - If you are analyzing HCP aging data you must generate a physio file in a similar format. This can be done using the provided function: createPhysio_fromHCP_aging.m. Input the folder directory with the physio data and run. This will work for the slicetiming of the HCP aging cohort. It has not been checked for the other cohorts.
3. Anatomical scan – used for registration (T1-weighted in the original work).

Pre-processing steps to be completed before running main_dataDrivenSeg.m:

1. Run FSL brain extraction on the fMRI data to generate a brain_mask.nii.gz file:
    a. The input_image.nii.gz should be 3D, so save an average of the 4D dataset or one volume.

```
CODE (FSL): bet input_image.nii.gz brain -m -f 0.4
```

2. Complete registration (complete the registrations however you prefer, but we use Ants for non-linear and fsl epi_reg for linear registrations; examples command lines are provided below).
    a. Register anatomical scan to MNI 152 1mm (non-linear registration, takes ~4 hours).

```
CODE (ANTS, replace MNI directories and T1_brain directories):
antsRegistration -d 3 -r
[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz ,1] -m
mattes[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz , 1 , 32, regular, 0.3 ] -t translation[ 0.1 ] -c
[10000x111110x11110,1.e-8,20] -s 4x2x1vox -f 6x4x2 -m
```

```
mattes[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz , 1 , 32, regular, 0.3 ] -t rigid[ 0.1 ] -c [10000x111110x11110,1.e-
8,20] -s 4x2x1vox -f 3x2x1 -m
mattes[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz , 1 , 32, regular, 0.3 ] -t affine[ 0.1 ] -c [10000x111110x11110,1.e-
8,20] -s 4x2x1vox -f 3x2x1 -m
mattes[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz , 0.5 , 32 ] -m
cc[ /N/soft/rhel8/fsl/6.0.5.2/data/standard/MNI152_T1_1mm_brain.nii.gz,
T1w_brain.nii.gz , 0.5 , 4 ] -t SyN[ .20, 3, 0 ] -c [ 100x100x50,-0.01,5 ] -s 1x0.5x0vox
-f 4x2x1 -u 1 -z 1 -o
[T1w_brain_FA_to_MNI152_T1_1mm_brain_ants,T1w_brain_FA_to_MNI152_T1_1mm_brain_ants_diff.n
ii.gz,T1w_brain_FA_to_MNI152_T1_1mm_brain_ants_inv.nii.gz]

antsApplyTransforms -d 3 -i T1w_brain.nii.gz -r
$FSLDIR/data/standard/MNI152_T1_1mm_brain.nii.gz -n linear -t
T1w_brain_FA_to_MNI152_T1_1mm_brain_ants1Warp.nii.gz -t
T1w_brain_FA_to_MNI152_T1_1mm_brain_ants0GenericAffine.mat -o
T1w_brain_FA_to_MNI152_T1_1mm_brain_ants_warped.nii.gz
```

b. Register anatomical scans to fMRI scans (linear registration).

CODE (FSL):
..In T1 weighted folder..

```
fslmaths T1w highres
fslmaths T1w_brain highres_head
```

..In fMRI folder..

```
epi_reg --epi=brain.nii.gz --t1=T1_path/highres_head --t1brain=T1_path/highres --
out=fmri2stru
convert_xfm -omat stru2fmri.mat -inverse fmri2stru.mat
```

3. Transform MNI 152 files into T1 space. (steps 2 and 3 can be combined if the transformation files are combined)
   a. UBA atlas to T1 space

CODE (ANTS):

```
antsApplyTransforms -e 3 -r folder_path/T1_brain.nii.gz -I
folder_path/UBA6bi_mni.nii.gz -t
[folder_path/T1_biascorr_brain_FA_to_MNI152_T1_1mm_brain_ants0GenericAffine.mat, 1] -t
folder_path/T1_biascorr_brain_FA_to_MNI152_T1_1mm_brain_ants1InverseWarp.nii.gz -o
folder_path/UBA6bi_stru.nii.gz -n NearestNeighbor
```

   b. SSS MNI segmentation to T1 space

CODE (ANTS):

```
antsApplyTransforms -e 3 -r folder_path/T1_brain.nii.gz -I
folder_path/MNI152_T1_1mm_SSS.nii.gz -t
[folder_path/T1_biascorr_brain_FA_to_MNI152_T1_1mm_brain_ants0GenericAffine.mat, 1] -t
folder_path/T1_biascorr_brain_FA_to_MNI152_T1_1mm_brain_ants1InverseWarp.nii.gz -o
folder_path/SSS_mask_stru.nii.gz -n NearestNeighbor
```

4. Transform MNI 152 files into fMRI space.
   a. UBA atlas to fMRI space – name the file UBA6bi_fmri.nii.gz

CODE (FSL):

```
flirt -in folder_path/UBA6bi_stru.nii.gz -ref folder_path/brain.nii.gz -out
folder_path/UBA6bi_fmri -applyxfm -init folder_path/stru2fmri.mat
```

   b. SSS MNI segmentation – name the file SSS_mask_fmri.nii.gz

CODE (FSL):

```
flirt -in folder_path/SSS_mask_stru.nii.gz -ref folder_path/brain.nii.gz -out
folder_path/SSS_mask_fmri -applyxfm -init folder_path/stru2fmri.mat
```

5. If the physio file is not in the CMRR format, generate a physio file (see Required local scan files #2)

Once the preprocessing steps 1-4 are complete and all the required files are in the fMRI folder, you can run **main_dataDrivenSeg.m –** This main script will call all the functions in the order that they should be completed. You will have to adjust the following variables: fMRI_scan_folder, fMRI_filename, physiology_file_name.

<u>All major code used to complete the segmentation:</u>

- main_dataDrivenSeg.m script – call this after all the registration and file transformations are completed. This main script will run the functions described below.

- readCMRRPhysio.m – This function reads the finger plethysmography data in the CMRR format (this script is provided from `https://github.com/CMRR-C2P/MB.git`)

- ppg_analysis.m – This function preprocessed the finger plethysmography data and outputs a quality control (QC) plot and metrics.

- align_fMRI_wholebrain.m – This function completes Step 1, described in the manuscript.

- meanwaveform_prelimArtery.m – This function completes Step 2, described in the manuscript.

- vesselMask_corrBased.m – This function completed Step 3, described in the manuscript.

- createPhysio_fromHCP_aging – this function will generate a physio file in the CMRR format for HCP aging participants.