Arterial Spin-Labeled MRI Preprocessing and Cerebral Blood Flow Computation

Arterial spin-labeled MRI images were preprocessed using *ASLPrep* 0.2.7beta, which is based on *Nipype* 1.6.1 (Gorgolewski et al. 2011).

Anatomical data preprocessing

sMRIPrep 0.6.1 was used to process the anatomical data. A total of 1 T1-weighted (T1w) image was found within the input BIDS dataset. The T1-weighted (T1w) image was corrected for intensity non-uniformity (INU) with `N4BiasFieldCorrection` (Tustison et al. 2010), which is distributed with *ANTs* 2.3.1 (Avants et al. 2008). *sMRIPrep* uses this T1w reference throughout the workflow. The T1w-reference was then skull-stripped with a *Nipype* implementation of the `antsBrainExtraction.sh` workflow using OASIS30ANTs as the target template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) was performed on the brain-extracted T1w reference image using *FSL*'s `FAST` (Zhang, Brady, and Smith 2001). Nonlinear registration of the brain-extracted T1w reference image to the brain-extracted template was accomplished using `antsRegistration`. The following template was selected for spatial normalization: *ICBM 152 Nonlinear Asymmetrical template version 2009c* (Fonov et al. 2011),

ASL data preprocessing

For the 1 ASL run found per subject (across all tasks and sessions), the following preprocessing was performed. First, the middle volume of the ASL timeseries was selected as the refernce volume and brain extracted using *Nipype*'s custom brain extraction workflow. Head-motion parameters were estimated using *FSL*'s `mcflirt` (Jenkinson et al. 2002). Next, ASLPrep wrote head-motion parameters to the ASL run's confound file.

Susceptibility distortion correction (SDC) was omitted. ASLPrep co-registered the ASL reference to the T1w reference using *FSL*'s `flirt` (Jenkinson and Smith 2001), which implemented the boundary-based registration cost-function (Greve and Fischl 2009). Co-registration used 6 degrees of freedom. The quality of co-registration and normalization to template was quantified using the Dice and Jaccard indices, the cross-correlation with the reference image, and the overlap between the ASL and reference images (e.g., image coverage). Several confounding timeseries were calculated, including both framewise displacement (FD) and DVARS. FD and DVARS are calculated using the implementations in in *Nipype* (following the definition by (Power et al. 2014)) for each ASL run. ASLPrep summarizes in-scanner motion as the mean framewise displacement and relative root-mean square displacement.

Cerebral blood flow computation and denoising

ASLPrep was configured to calculate cerebral blood flow (CBF) using the following methods.

The cerebral blood flow (CBF) was quantified from preprocessed ASL data using a general kinetic model (Detre et al. 1992; Alsop et al. 2015).

Structural Correlation based Outlier Rejection (SCORE) algorithm was applied to the CBF timeseries to discard CBF volumes with outlying values (Dolui et al. 2017) before computing the mean CBF. Following SCORE, the Structural Correlation with RobUst Bayesian (SCRUB) algorithm was applied to the CBF maps using structural tissue probability maps to reweight the mean CBF (Dolui et al. 2017; Dolui, Wolk, et al. 2016).

CBF was also computed with Bayesian Inference for Arterial Spin Labeling (BASIL) (Chappell et al. 2009), as implemented in *FSL* 6.0.3. BASIL computes CBF using a spatial regularization of the estimated perfusion image and additionally calculates a partial-volume corrected CBF image (Chappell et al. 2011). For each CBF map, the ROIs for the following atlases were extracted: the Harvard-Oxford and the Schaefer 200 and 400-parcel resolution atlases.

The Quality evaluation index (QEI) was computed for each CBF map (Dolui, Wolf, et al. 2016). QEI is based on the similarity between the CBF and the structural images, the spatial variability of the CBF image, and the percentage of grey matter voxels containing negative CBF values.
All resampling in *ASLPrep* uses a single interpolation step that concatenates all transformations. Gridded (volumetric) resampling was performed using `antsApplyTransforms`, configured with *Lanczos* interpolation to minimize the smoothing effects of other kernels (Lanczos 1964). Many internal operations of *ASLPrep* use *Nilearn* 0.8.0 (Abraham et al. 2014), *NumPy* (Harris et al. 2020), and *SciPy* (Virtanen et al. 2020). For more details of the pipeline, see [the *ASLPrep* documentation.].

Copyright Waiver

References