REPORT Documentation

Release V1

DIMUTHU

Jul 25, 2020

Contents

1 Installation **3**
- 1.1 Install Singularity . . . 3
- 1.2 Running prepdwi . . . 3

2 Usage **7**
- 2.1 BIDS requirements . . . 8
- 2.2 Flag description . . . 8
- 2.3 Participant Level . . . 9
- 2.4 Group Level . . . 10
- 2.5 Participant2 Level . . . 10

3 Prepdwi Pipeline **11**

4 Support **13**

5 Citing Prepdwi **15**
- 5.1 References: . . . 16

Index **17**

Prepdwi is a BIDS App developed by the Khan Lab to pre-process diffusion-weighted MRI data, and optionally also perform simple probabalistic tractography. Steps to generate a pre-processed DWI image include de-noising, unringing, EPI distortion correction (top-up, and if necessary non-linear image registration), eddy correction, gradient correction (requires coefficient file), and rigid registration to T1w. The app also generates FSL FDT dtifit maps, DKE kurtosis maps (for multi-shell DWI only), and FSL BEDPOST pre-processing.

CHAPTER 1

Installation

Prepdwi is a BIDS App, and is thus built into a Docker/Singularity container. Our current build process uses Docker to build and test with continuous integration, Docker Hub to store the latest version and each release, and copies of these containers are also stored on Singularity Hub. The *latest* tag in both Docker Hub and Singularity Hub is the latest development version of the pipeline, with the releases tagged with v0.* (latest release is v0.0.8 as of September 4, 2018).

https://hub.docker.com/r/khanlab/prepdwi/ https://singularity-hub.org/collections/392

You can run prepdwi using either Docker or Singularity. We recommend Singularity if you are running the pipeline on a shared compute system. For instructions on running BIDS Apps with Docker, please see http://bids-apps.neuroimaging.io

1.1 Install Singularity

To run a singularity image, first you need to have singularity installed in your local computer or server. To install singularity, please follow the instructions in the singularity website.

https://www.sylabs.io/guides/2.5.1/user-guide/

1.2 Running prepdwi

Once Singularity set up on your computer you can start running prepdwi. The following is an example of running prepdwi on a BIDS dataset.

1.2.1 Example

For this example, let's assume that your downloaded singularity image is saved in the path home/singularity/Singularity.0.07g and your BIDS data is saved in home/data/bids. you can rename the image name from Singularity.0.07g to prepdwi_7g for simplicity

The data in the "bids" folder should be in the BIDS format. For more details about BIDS format, read: http://bids.neuroimaging.io/

participant level analysis

In prepdwi, there are several levels of analysis named as participant, group and participant2. To learn more about these analysis levels and optional arguments/flags, read (put a link here)

The basic structure for running prepdwi is:

```
singularity run <path_to_prepdwi_image> <bids_dir> <output_dir> {participant,group,
↪participant2} <optional arguments>
```

For our example create a directory to save the output inside the project directory as "derrivatives".

At the command line type:

```
singularity run home/singularity/prepdwi_version.img home/project/bids home/project/
↪derrivatives participant
```

Or, if you have access to Khanlab Graham server and have neuroglia helpers. installed you can submit it as a job.

```
bidsBatch prepdwi_version <bids_dir> <output_dir> participant
```

This will run the participant level analysis for all the subjects in the bids folder and will save the resultst to the derrivatives folder. This code is running the FSL BEDPOSTX. which takes a long time and we highly recommend using highspeed computer server to run this code. (Typical time is 24 hours in Sharcnet) Once the process is completed, you will see a "work" folder and a "prepdwi" folder inside the "derrivatives" directory. To learn what to expect inside thses folders, read (link to the cookbook)

group level analysis

After running the participant level analysis, you can run a group level analysis to see how good the registarions are. To know more about the group level analysis, read (link)

At the command line type:

```
singularity run home/singularity/prepdwi_version.img home/project/bids home/project/
↪derrivatives group
```

Or, for Khanlab members

```
bidsBatch prepdwi_version <bids_dir> <output_dir> group
```

IMPORTANT: Makesure that the ourput directory is the same as the one for "participant" level.

Once the group level analysis is completed, you will see a new folder inside the "derrivatives" directory called "reports". There you will see a list of html files for each subject which shows the qulaity of the registration at each process. The failed registrations can be identified if the red contour plots are not overlapping with the template image. For the registration failed cases, you can re-run prepdwi participant level using –reg_init_participant flag which is explained in the (link to cookbook).

participant2 level analysis

If the participant1 level is completed you can run participant2 level analysis on the data. To know more about participant2 level, read (link)

At the command line type:

```
singularity run home/singularity/prepdwi_version.img home/project/bids home/project/
↪derrivatives participant2
```

Or, for Khanlab members

```
bidsBatch prepdwi_version <bids_dir> <output_dir> participant2
```

IMPORTANT: Makesure that the ourput directory is the same as the one for "participant" level.

Once the participant2 level analysis is completed, you will see a new folder inside the "derrivatives" directory called "bedpost". Also you will see several csv files for connectivity matrix and FA matrices.

CHAPTER 2

Usage

The prepdwi pipeline is a BIDS App that has three different levels of analysis: a pre-processing pipeline (*participant*), tractography (*participant2*), and QC report generation (*group*).

This page describes the command-line arguments for prepdwi and for each level of analysis. For details on the pipeline itself, please see *Prepdwi Pipeline*.

usage:

```
prepdwi bids_dir output_dir {participant,group,participant2} <optional arguments>
          [--participant_label PARTICIPANT_LABEL [PARTICIPANT_LABEL...]]
          [-w WORK_DIR]  (scratch directory)
          [--n_cpus NCPUS] (for bedpost, default: 8)

 participant options:
          [--matching_dwi MATCHING_PATTERN]
          [--matching_T1w MATCHING_STRING]
          [--reg_init_participant PARTICIPANT_LABEL]
          [--grad_coeff_file GRAD_COEFF_FILE]

          [--no-regT1]
          [--no-topup]
          [--no-bedpost]
          [--no-dke]

 participant2 (probtrack connectivity) options:
          [--nprobseeds] N (for probtrackx, default: 5000)
      Choose built-in atlas:
          [--atlas NAME (default: dosenbach)

       Available built-in atlas labels/csv:
          cort_striatum_midbrain        dosenbach  yeo17  yeo17_striatum  yeo7  yeo7_
↪striatum

       Customize atlas labels:
          [--atlas_space NAME (MNI152_1mm or MNI152NLin2009cAsym)]
```

(continues on next page)

(continued from previous page)

```
          [--atlas_label_nii NIFTI]
          [--atlas_label_csv LABEL_INDEX_CSV]
```

2.1 BIDS requirements

The prepdwi pipeline requires at least one `dwi` image (`nii`/`nii.gz`), in the `dwi` folder as per the BIDS standard. A `json` sidecar is also required for each DWI nifti, with the `PhaseEncodingDirection` and `EffectiveEchoSpacing` variables set. A `T1w` nifti (in the `anat` subfolder) is also highly recommended, but not required if the `--no-regT1` flag is used. If multiple T1w nifti images exist, only the first one (from alphanumeric sorting) will be used. To specify a particular T1w image to use, you can use the `--matching-T1w` option.

2.2 Flag description

2.2.1 General flags

Flag	Options	Description
–participant_label	PARTICIPANT_LABEL	Runs for a subset of subjects
-w	WORK_DIR	Specify scratch folder for temporary files
–n_cpus NCPUS.		Number of CPUs to use for parallel steps (e.g. BEDPOST with GNU parallel). Default: 8

2.2.2 Participant level flags

Flag	Options	Description
–matching_dwi	MATCHING_PATTERN	
–matching_T1w	MATCHING_STRING	
–reg_init_participant	PARTICIPANT_LABEL	Use this flag to re-run participant level for the subjects with failed registrartoins observed in the group level. Use a subject with good registration to initialize the registartion. The subject ID should match the ID as in the work folder.
–grad_coeff_file	GRAD_COEFF_FILE	Use the provided gradient coeffient file (Siemens format) for gradient non-linearity correction (disabled otherwise)
–no-regT1		Disable rigid registration and resampling of processed DWI to the T1 space
–no-topup		Disable Topup correction (use this if opposite phase encode runs do not exist)
–no-bedpost		Disable BEDPOST pre-processing
–no-dke		Disable DKE Kurtosis map estimation

2.2.3 Participant2 level flags

Flag	Options	Description
–nprobseeds	N	Number of seeds for probtrackx, default: 5000
–atlas[1]	dosenbach	Name of the atlas to use for connectivity. default: dosenbach
	cort_striatum_midbrain	
	dosenbach yeo17	
	yeo17_striatum	
	yeo7	
	yeo7_striatum	
–atlas_space[2]	MNI152_1mm	MNI space where the atlas is defined
	MNI152NLin2009cAsym	
–atlas_label_nii[2]	NIFTI	Nifti file with labeled ROIs
–atlas_label_csv[2]	LABEL_INDEX_CSV	CSV file with indices and labels (each line as: Label_Name,Label_Number)

[1] flgas for available atlases, [2] flags for user defined atlases

2.3 Participant Level

At the participant level the T1 data and DWI data are being pre-processed using the following steps. At the end of preprocessing, it will run FSL DTI fit and generates fractional anisotropy (FA) images and mean diffusivity images and other results from DTI fitting, which can be found in the prepdwi folder.

2.3.1 Denoising and Unringing

Denoising is performed as the first step of our pipeline. We are using the dwidenoise tool in MRtrix pipeline to perform the denoising process. After running denoisining the unring tool was used to remove the Gibbs ringing artefact in images.

2.3.2 top-up

topup is a FSL tool used for estimating and correcting susceptibility induced distortions in our pipeline. This can be used only if you have reversed phase-encoded pairs of images with distortion going in opposite directions. If not, use –no-topup flag.

2.3.3 Eddy Current Correction

After running topup, the images are corrected for Eddy_Current using the eddy tool in FSL.

2.3.4 T1w-T1w template (MNI152_1mm and MNI152NLin2009cAsym) registration

2.3.5 BEDPOST (optional)

FSL BEDPOSTX performs Markov Chain Monte Carlo sampling to build up distributions on diffusion parameters at each voxel. It creates all the files necessary for running probabilistic tractography. To learn more information about the BEDPOSTX, please visit the FSL webpage. The results from the BEDPOSTX can be found at the bedpost folder created after running the participant level.

2.4 Group Level

At the Group Level analysis, prepdwi reads all the processed data in participant level and creates a qulaity report for each subject showing how good the registrations are. You can't run group level for a single subject. Once the group level analysis is completed, you will see a new folder inside the "derrivatives" directory called "reports". There you will see a list of html files for each subject which shows the qulaity of the registration at each process. The failed registrations can be identified if the red contour plots are not overlapping with the template image. For the registration failed cases, you can re-run prepdwi participant level using –reg_init_participant flag.

To use the –reg_init_participant flag, you have to pick a subject which has a successful good registration. Then Prepwi will use that as the initial image to register the images of the subjects you want.

```
singularity run home/singularity/prepdwi_version.img home/project/bids home/project/
↪derrivatives participant --reg_init_participant <subj-ID>
```

Or, for Khanlab members

```
bidsBatch prepdwi_version <bids_dir> <output_dir> participant --reg_init_participant
↪<subj-ID>
```

Here the subject ID should be as same as in the work folder. Not as in the bids folder. If there are multimple session for a subject, the session name will be added as a suffix to the subject ID in the work folder. Therefore you have to use the subject ID as it is in the work folder.

2.5 Participant2 Level

Runs probtrackx network connectivity between all regions in a given atlas labels file. Uses either can use atlases with the –atlas option, where predefined atlases are defined in the cfg folder; or can specify a new atlas with the –atlas_* options. Participant2 level generates a connectivity matrix of the averaged fiber density between each ROI in the atlas. This matrix can be found in the prepdwi folder as a csv file. The fiber desity is calculated by averaging the number of connections from each seed volxel to a give target by the number of seeds (default: 5000). Then it is averaged by the number of voxols per ROI.

CHAPTER 3

Prepdwi Pipeline

The prepdwi pipeline is a BIDS App that has three different levels of analysis: a pre-processing pipeline (*participant*), tractography (*participant2*), and QC report generation (*group*).

The pre-processing pipeline aims to produce a filtered and corrected diffusion-weighted image, along with a set of basic quantitative maps, that can then be used for downstream analyses (fitting, tractography, voxel-based analysis, microstructural modelling, and so on). The app is written in BASH, with each step generally written as a separate script, each called by the main prepdwi script. A summary of the processing steps is provided here, along with references to the corresponding code.

T1 pre-processing (code)

- Skull-stripping with FSL BET (options: -f 0.4 -B) [cite]
- Non-uniformity correction with ANTS N4BiasFieldCorrection [cite]
- intensity normalization by mean intensity

T1 atlas registration

- Non-linear registration is performed to obtain transformations to/from the MNI152_1mm and MNI152NLin2009cAsym templates, and transform labels to the subject T1 space
- The pipeline will perform this step on all templates in the atlases folder, using the `t1/t1.brain.inorm.nii.gz` image to register, and transforming the labels in `labels/<label_names>` folders
- Affine registration is performed with reg_aladin (NiftyReg 1.3.9) code
- Non-linear (b-spline) registration is performed with reg_f3d (NiftyReg 1.3.9) code
- Atlas labels from the `labels` folders are transformed to the subject T1 space with reg_resample (NiftyReg 1.3.9) code

DWI pre-processing:

- DWI Denoising is performed on the raw DWI using dwidenoise (MRtrix3) code
- Removal of Gibb's ringing artifacts was performed with the unring tool code
- **If multiple phase-encode directions exist:**

  - Susceptibility-induced distortions are corrected with top-up & eddy (FSL)
  - Uses the b02b0 preset on average b0 images and requires JSON flags `PhaseEncodingDirection` and `EffectiveEchoSpacing`, and uses the voxel dimension in the phase encode direction to generate the `acqp` file code)
  - Performs eddy-current correction using eddy_openmp, concatenates all scans, –repol to replace outliers code)

- **If multiple phase-encode directions do NOT exist:**
  - Performs eddy-current correction using eddy_openmp, concatenates all scans, –repol to replace outliers code)
  - Performs non-linear registration (NiftyReg) for EPI distortion correction (uses skull-stripped T1 with inverted intensities, rigidly-registered to the avgB0 as a reference; uses b-spline registration (reg_f3d, -be 0.001), and warps DWI code)
- Rigid alignment to T1 space (reg_aladin, rotates bvecs, resamples to either specified resolution, or original dwi resolution code)
- **If a gradient coefficient file exists:**
  - Performs gradient unwarping (cubic interp, jacobian modulation, generates grad_dev file, concatenates with DWI-T1 rigid transform code)
- Generates mean DWI for each shell (code)
- FSL BEDPOST
- **if (multi-shell)**
  - DKE fitting
- export to BIDS-like output

etc:

- BIDS [cite]
- BIDS-Apps [cite]
- Docker [cite]
- Singularity [cite]
- neuroglia-core/dwi [cite]

Built-in external atlases:

- Dosenbach [cite]
- Yeo7, Yeo17 [cite]

CHAPTER 4

Support

The easiest way to get help with the project is to open an issue on Github.

You can also e mail alik@robarts.ca for support.

CHAPTER 5

Citing Prepdwi

All structural and diffusion data were processed and analyzed using an open-source and containerized application, `prepdwi`, which uses the BIDS [1] and BIDS Apps [2] standards to perform standardized pre-processing, fitting, image registration, and tractography.

T1w images were skull-stripped using `bet` (FSL, using the `-f 0.4 -B` options, [3]), corrected for non-uniformities `N4BiasFieldCorrection` (ANTS, [4]), and normalized by the mean intensity within the brain mask. Pre-processed T1w images were registered with standard templates (MNI ICBM152 non-linear 6th generation symmetric template, referred to as `MNI152_1mm` in FSL, and the `MNI152NLin2009cAsym` template) using an initial affine registration using block-matching [5], followed by deformable b-spline registration [6], both implemented in NiftyReg v1.3.9. Overlay visualizations depicting the skull-stripping, affine registration, and deformable registration were generated for each subject to check for failures. Failures in affine registration could occur, but were corrected by forcing initialization with an existing transformation matrix. Discrete and probabilistic segmentation images in the template spaces were automatically propagated to each subject's T1w space, using nearest neighbour interpolation for discrete segmentations, and linear for probabilistic segmentations.

Diffusion-weighted MRI data were pre-processed with denoising using a local PCA method with (`dwidenoise` from mrtrix3, [7]), and correction of ringing artifacts with the `unring` tool [8]. Eddy current distortions were corrected using `eddy` (FSL, [9]), with the `--repol` option enabled for outlier replacement [10]. If multiple phase-encoding polarities were used in the acquisition, `top-up` [11]was used to correct for susceptibility distortions, with the resulting parameters fed into eddy. If data were not sufficient to run `top-up`, a registration-based susceptibility distortion correction was performed following `eddy`, using B-spline deformable registration [6] between the average b0 image and a T1w volume with inverted intensities. Finally, within-subject rigid registration of the corrected DWI volume and the T1w volume was performed using block-matching[5], to bring the DWI images in the same space as the T1w, where atlas labels were also propagated. If gradient non-linearities were provided through spherical harmonic coefficients (e.g. for the AC84 gradient system on the 7T), these were used with the `gradient_unwarp` tool [12] to generate a non-linear transformation, which was composed with the T1w linear transformation to resample the DWI images into the corrected T1w space in a single step. Modulation with the determinant of the Jacobian of the unwarping was used to correct for intensity differences in the magnitude images due to gradient non-linearities. Pre-processed DWI images in the T1w space were then used to estimate diffusion tensor metrics using `dtifit` (FSL, [13]), and ball and stick modelling for probablistic tractography using `bedpostx` [14]. If multi-shell diffusion data was detected, diffusion kurtosis metrics were computed using the DKE toolbox [15].

5.1 References:

1. Gorgolewski KJ, Auer T, Calhoun VD, Craddock RC, Das S, Duff EP, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Sci Data. 2016;3: 160044. doi:10.1038/sdata.2016.44
2. Gorgolewski KJ, Alfaro-Almagro F, Auer T, Bellec P, Capotă M, Chakravarty MM, et al. BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. PLoS Comput Biol. 2017;13: e1005209. doi:10.1371/journal.pcbi.1005209
3. Smith SM. Fast robust automated brain extraction. Hum Brain Mapp. 2002;17: 143–155. doi:10.1002/hbm.10062
4. Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, et al. N4ITK: improved N3 bias correction. IEEE Trans Med Imaging. 2010;29: 1310–1320. doi:10.1109/TMI.2010.2046908
5. Modat M, Cash DM, Daga P, Winston GP, Duncan JS, Ourselin S. Global image registration using a symmetric block-matching approach. J Med Imaging (Bellingham). 2014;1: 024003. doi:10.1117/1.JMI.1.2.024003
6. Modat M, Ridgway GR, Taylor ZA, Lehmann M, Barnes J, Hawkes DJ, et al. Fast free-form deformation using graphics processing units. Comput Methods Programs Biomed. 2010;98: 278–284. doi:10.1016/j.cmpb.2009.09.002
7. Veraart J, Novikov DS, Christiaens D, Ades-Aron B, Sijbers J, Fieremans E. Denoising of diffusion MRI using random matrix theory. Neuroimage. 2016;142: 394–406. doi:10.1016/j.neuroimage.2016.08.016
8. Kellner E, Dhital B, Kiselev VG, Reisert M. Gibbs-ringing artifact removal based on local subvoxel-shifts. Magn Reson Med. 2016;76: 1574–1581. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26054
9. Andersson JLR, Sotiropoulos SN. An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging. Neuroimage. 2016;125: 1063–1078. doi:10.1016/j.neuroimage.2015.10.019
10. Andersson JLR, Graham MS, Zsoldos E, Sotiropoulos SN. Incorporating outlier detection and replacement into a non-parametric framework for movement and distortion correction of diffusion MR images. Neuroimage. 2016;141: 556–572. doi:10.1016/j.neuroimage.2016.06.058
11. Andersson JLR, Skare S, Ashburner J. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. Neuroimage. 2003;20: 870–888. doi:10.1016/S1053-8119(03)00336-7
12. Jovicich J, Czanner S, Greve D, Haley E, van der Kouwe A, Gollub R, et al. Reliability in multi-site structural MRI studies: effects of gradient non-linearity correction on phantom and human data. Neuroimage. 2006;30: 436–443. doi:10.1016/j.neuroimage.2005.09.046
13. Jenkinson M, Beckmann CF, Behrens TEJ, Woolrich MW, Smith SM. FSL. Neuroimage. 2012;62: 782–790. doi:10.1016/j.neuroimage.2011.09.015
14. Behrens TEJ, Woolrich MW, Jenkinson M, Johansen-Berg H, Nunes RG, Clare S, et al. Characterization and propagation of uncertainty in diffusion-weighted MR imaging. Magn Reson Med. 2003;50: 1077–1088. doi:10.1002/mrm.10609
15. Tabesh A, Jensen JH, Ardekani BA, Helpern JA. Estimation of tensors and tensor-derived measures in diffusional kurtosis imaging. Magn Reson Med. 2011;65: 823–836. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.22655

#versions: neuroglia-dwi:latest mrtrix 3.0_RC3 camino 2019-02-01 1c4ef77615d103d43adcff6c79b72d0bbdac0897 unring 2017-02-17 dke v1.0 niftyreg 1.3.9 fsl v6.0 (fslinstaller 3.0.12)

Index

S

Syntax
    TOC Tree, 5, 10, 12, 16

T

TOC Tree
    Syntax, 5, 10, 12, 16