

Jan 14, 2026

A Reproducible Diffusion MRI Processing Pipeline for Microstructure and Glymphatic Analysis Using Neurodesk Containers

DOI

<https://dx.doi.org/10.17504/protocols.io.3byl48eezvo5/v1>

Yakubu Kanyiri Ahmed¹, Ayebare Vicky², Seth Kyei Kwabena Kukudabi³, Jeffrey Gameli Amlalo⁴, Claudia Takyi Ankomah⁵, Sulaiman Sunusi Sulaiman⁶, Harrison Aduluwa⁷, Benjamin Anyanwu⁸, Udunna Anazodo^{7,9}, Abdalla Z Mohamed¹⁰, Morgan Hough¹¹, Cristian Montalba¹², Ethan Draper¹³, CAMERA: democratizing MRI¹⁴

¹Komfo Anokye Teaching Hospital, Kumasi, Ghana;

²Mbarara University of Science and Technology, Mbarara, Uganda;

³University for Development Studies, Tamale, Ghana; ⁴University of Cape Coast, Cape Coast, Ghana;

⁵Kwame Nkrumah University of Science and Technology, Kumasi, Ghana;

⁶Federal Medical Centre Birnin Kudu, Birnin Kudu, Nigeria;

⁷Montreal Neurological Institute, McGill University, Montreal, Canada;

⁸Regions Healthcare Hospitals and Specialist Clinics, Mgbirichi Ohaji, Imo State, Nigeria;

⁹Medical Artificial Intelligence Lab, Lagos, Nigeria;

¹⁰United Arab Emirates University, Al Ain, Abu Dhabi, United Arab Emirates;

¹¹NeuroTechX & Biopunk Lab, San Francisco, USA;

¹²Biomedical Imaging Center, Pontificia Universidad Catolica de Chile;

¹³Department of Bioengineering, Imperial College London, UK;

¹⁴Consortium for Advancement of MRI Education and Research in Africa

CONNExIN Microstructure



YAKUBU KANYIRI AHMED

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account



DOI: <https://dx.doi.org/10.17504/protocols.io.3byl48eezvo5/v1>

Protocol Citation: Yakubu Kanyiri Ahmed, Ayebare Vicky, Seth Kyei Kwabena Kukudabi, Jeffrey Gameli Amlalo, Claudia Takyi Ankomah, Sulaiman Sunusi Sulaiman, Harrison Aduluwa, Benjamin Anyanwu, Udunna Anazodo, Abdalla Z Mohamed, Morgan Hough, Cristian Montalba, Ethan Draper, CAMERA: democratizing MRI 2026. A Reproducible Diffusion MRI Processing Pipeline for Microstructure and Glymphatic Analysis Using Neurodesk Containers. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.3byl48eezvo5/v1>

License: This is an open access protocol distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working.

Created: December 15, 2025

Last Modified: January 14, 2026

Protocol Integer ID: 235017

Keywords: Diffusion MRI, Glymphatic System, 1.5 Tesla, Africa, Microstructure, reproducible diffusion mri processing pipeline for microstructure, reproducible diffusion mri processing pipeline, standardized diffusion mri preprocessing, end pipeline for diffusion mri processing, diffusion mri processing, research diffusion mri data, preprocessed diffusion dataset, white matter hyperintensity segmentation, perivascular diffusion characteristic, diffusion dataset, mri, quantitative assessment of white matter microstructure, white matter hyperintensity mask, containerized tools within the neurodesk environment, white matter microstructure, neurodesk environment, microstructural reconstruction, using neurodesk container, glymphatic assessment, microstructural reconstructions such as fiber orientation distribution, including image quality metric, neurodesk containers this protocol, glymphatic analysis, glymphatic assessment through bid, diffusivity component, image quality metric

Disclaimer

This pipeline was created by the 2025 Microstructure Team as part of a collaborative project focused on establishing a reproducible workflow for diffusion MRI processing, microstructural reconstruction, and glymphatic assessment using Neurodesk containers.

The content provided herein is for academic and technical research purposes only and does not serve as clinical, medical, or legal advice. This protocol has not been evaluated or approved by any regulatory agency for clinical diagnostic use. While this pipeline offers a standardized, BIDS-compliant approach to quantifying biomarkers such as the DTI-ALPS index and fiber orientation distributions, it is not a mandatory or clinically validated tool.

Implementation of this pipeline and the subsequent interpretation of resulting biomarkers are the sole responsibility of the user. The authors and their affiliated organizations assume no liability for the accuracy of findings in clinical settings or for any use or misuse of the pipeline's content. Users are responsible for conducting their own objective research and ensuring compliance with local Institutional Review Board (IRB) and ethical guidelines regarding participant data security and privacy.

Abstract

This protocol describes an accessible and reproducible end-to-end pipeline for diffusion MRI processing and microstructure analysis using containerized tools within the Neurodesk environment on Google Colab. It enables standardized diffusion MRI preprocessing, microstructural reconstruction, white matter hyperintensity segmentation, and glymphatic assessment through BIDS-compliant workflows designed for clinical and research diffusion MRI data, including 1.5 T acquisitions. Successful execution yields organized derivatives including image quality metrics, anatomically preprocessed structural data, preprocessed diffusion datasets, microstructural reconstructions such as fiber orientation distributions or tensor-based measures, and white matter hyperintensity masks. The pipeline also produces diffusivity components and ALPS index outputs that support quantitative assessment of white matter microstructure and perivascular diffusion characteristics in a reproducible analysis framework.



Before start

This section will help you prepare with the right materials needed to successfully implement this protocol.

Computational Infrastructure: Neurodesk and Google Colab

To ensure full reproducibility, this pipeline is designed to run within **Neurodesk**, a scalable data analysis platform for reproducible neuroimaging. By utilizing Neurodesk Singularity containers on **Google Colab**, researchers can access standardized software environments without the need for complex local installations. This approach prevents "dependency hell" and ensures that results remain consistent across different hardware configurations and timeframes.

Hardware and Environment Setup

- Google Colab environment (typically ~25GB RAM)
- **Container Runtime:** Apptainer/ Singularity
- **Software Distribution:** CernVM File System (CVMFS).

Initial Environment Deployment

The deployment of a stable, containerized environment is the first prerequisite for pipeline reproducibility. We utilize the Neurocommand framework to mount the global software repository.

1. Open a Google Colab notebook and select GPU runtime (optional but recommended for TRUENET/QSIPrep stages)
2. Initialize the CVMFS client and Apptainer runtime to access Neurodesk modules.



Command

System Initialization

```
import os
os.environ[LD_PRELOAD] = ;
os.environ[APPTAINER_BINDPATH] = /content
os.environ[MPLCONFIGDIR] = /content/matplotlib-mpldir
os.environ[LMOD_CMD] = /usr/share/lmod/lmod/libexec/lmod

!curl -J -O
https://raw.githubusercontent.com/neurodesk/neurocommand/main/googlecolab_setup.sh
!chmod +x googlecolab_setup.sh
!./googlecolab_setup.sh

os.environ[MODULEPATH] = :.join(map(str, list(map(lambda x:
os.path.join(os.path.abspath(/cvmfs/neurodesk.ardc.edu.au/neurodesk-
modules/), x),os.listdir(/cvmfs/neurodesk.ardc.edu.au/neurodesk-modules/)))))
```

Checkpoint: Verify the successful mount by checking for the existence of the Neurodesk applications/modules like this:

```
import lmod
await lmod.avail()
```

and use the applications/modules like this:

```
await lmod.load('fsl/6.0.4')
!bet
```

Mount Drive

Before you continue, you need to mount your drive on colab to be able to continue with the rest of the pipeline. Run this command below to mount your drive.



Command

Drive Mount

```
from google.colab import drive
drive.mount(/content/drive)
```

Directory Structure Setup

A hierarchical directory structure is required for BIDS compliance and organized derivatives. Define your project root and create the necessary subfolders for raw and processed data.

Command

Creating Project Directories

```
# Define your root directory
!export PROJECT_DIR=/content/drive/MyDrive/<YOUR_PROJECT_NAME>

!mkdir -p $PROJECT_DIR/sourcedata
!mkdir -p $PROJECT_DIR/rawdata
!mkdir -p $PROJECT_DIR/derivatives/mriqc
!mkdir -p $PROJECT_DIR/derivatives/fmriprip
!mkdir -p $PROJECT_DIR/derivatives/truenet
!mkdir -p $PROJECT_DIR/derivatives/qsiprep
!mkdir -p $PROJECT_DIR/derivatives/qsirecon
!mkdir -p $PROJECT_DIR/derivatives/dti_alps
```

Checkpoint: Verify that the `$PROJECT_DIR` contains the *sourcedata*, *rawdata*, and *derivatives* folders.

<code>\$PROJECT_DIR/</code>	# Root project folder
— sourcedata/	# Raw scanner data (Original DICOMs)
— rawdata/	# BIDS-formatted dataset (NIfTI)

└─ dataset_description/	# JSON file with study-level metadata
└─ participants.tsv	# Table containing subject demographics
└─ sub-[ID]/	# Subject-specific raw data (anat, dwi, fmap)
└─ derivatives/	# All generated analysis outputs
└─ mriqc/	# Visual and statistical quality control reports
└─ fmriprep/	# Preprocessed structural and surface-normalized data
└─ qsiprep/	# Preprocessed diffusion data (denoised and unwarped)
└─ qsirecon/	# Reconstruction outputs (FODs, Connectivity matrices)
└─ truenet/	# Deep-learning White Matter Hyperintensity (WMH) masks
└─ dti_alps/	# Parallel workspace for Glymphatic Index calculation
└─ work/	# Temporary files used during pipeline execution

Software Requirement and Containers

This pipeline utilizes containerized modules within Neurodesk.

	A	B	C	D
	Tool Category	Container/Module	Version	Primary Purpose
	Data Organization	BIDScoin	4.6.2	DICOM to BIDS conversion
	Quality Control	MRQC	24.0.2	Image Quality Metrics
	Structural Preprocessing	fMRIPrep	25.1.3	Anatomical preprocessing
	Diffusion Preprocessing	QSIprep	1.0.1	Diffusion preprocessing
	Micromodeling	QSIRecon	1.1.0	Microstructural Modeling (FOD)
	Segmentation	FSL	6.0.7.18	WMH segmentation
	Surface Analysis	FreeSurfer	8.1.0	Surface sampling and reconstruction

	A	B	C	D
	ALPS	ANTS, AFNI, Mrtrix3	2.6.0, 25.2.03, 3.0.4	ALPS index calculation

Note

While specific versions were used for validation of this pipeline, users should verify the latest stable releases. It is recommended to check for the latest stable releases to ensure compatibility with the current Neurodesk environment.

Use `await lmod.avail()` after initializing neurodesk on colab to list currently available software versions in the Neurodesk repository.

DATASET

- 1 This protocol was optimized using a small (14 young adults subjects) Nigerian normative sample dataset locally collected using a United Imaging Health 1.5 Tesla MRI System. The dataset includes T1w and T2 FLAIR volumes. The Diffusion Weighted Imaging (DWI) data is a single shell data with 1 b_0 volume and 64 directions $b=1000 \text{ s/mm}^2$. $b=0$ volume was collected with each subject. Two subjects included reversed phase-encoding volumes (AP/PA) that improve preprocessing.

DATA CONVERSION AND BIDS STANDARDIZATION

- 2 Consistent data organization is essential for automated workflows¹. We use BIDScoin, which utilizes a template-based approach to convert DICOMs into a BIDS-compliant format².
- 2.1 Upload raw DICOM files to a designated `$SOURCE` directory on Google Drive or the local Colab disk.
- 2.2 Use *bidsmapper* to scan headers and generate a YAML mapping file. This file is crucial for telling bidscoiner how to convert your DICOM data into the BIDS format.

Command

Scans raw DICOM headers and generates a template YAML mapping file to define BIDS entities.

bidsmapper

```
await lmod.load(bidscoin/4.6.2)
```

```
#Scan DICOMs and generate the initial mapping
!bidsmapper $SOURCE_DICOM_DIR $BIDS_DIR
```

- 2.3 Adjust the mapping in the BIDS Editor GUI (forwarded via X11 if using a local Neurodesk instance or edited manually in the YAML file in Colab). It was done manually in our case.

Luckily, bidsmapper already uses a default template, which you can copy and then modify. Use this command to copy the default template into your \$BIDS_DIR

Command

This command copies the template bidsmap.yaml file into your \$BIDS_DIR for you to edit manually

Copy

```
import os
import shutil

# Define the source template bidsmap and the target location
TEMPLATE_BIDS_MAP = /root/.bidscoin/4.6.2/templates/bidsmap_dccn.yaml
BIDSMAP_DIR = os.path.join(BIDS_DIR, code/bidscoin)
BIDSMAP_FILE = os.path.join(BIDSMAP_DIR, bidsmap.yaml)

# Create the target directory if it doesnt exist
os.makedirs(BIDSMAP_DIR, exist_ok=True)

# Copy the template bidsmap to the target location
shutil.copy(TEMPLATE_BIDS_MAP, BIDSMAP_FILE)

print(fCopied template bidsmap to: {BIDSMAP_FILE})
```

Now that you have a bidsmap.yaml file at \$BIDSMAP_DIR, you need to edit it to accurately reflect your DICOM series and how they should be mapped to BIDS. You can do this by:

1. **Opening the file directly in Google Colab:** Navigate to the file browser on the left sidebar, find the \$BIDSMAP_DIR, and double-click to open it.
2. **Downloading the file and editing it locally:** You can download the file, edit it with a text editor on your computer, and then re-upload it to the same location.

Ensure the following mappings:

- *[Modality]* → *[BIDS_Entity]/sub-[LABEL]_[Suffix].nii.gz* (e.g., anat/sub-xx_T1w.nii.gz, dwi/sub-xx_dwi.nii.gz).

You'll need to inspect your DICOM data (e.g., using a DICOM viewer or by checking the headers) to understand the values that need to be matched in the bidsmap.yaml file.

Once you have edited and saved the bidsmap.yaml file, you can proceed to run the bidscoiner command, as it should find the mapping file.

Command

Executes the conversion of source DICOM data into NIfTI files and JSON sidecars based on the mapping file.

bidscoiner

```
# Execute the conversion based on the mapping
!bidscoiner $SOURCE_DICOM_DIR $BIDS_DIR
```

Checkpoint: check \$BIDS_DIR to see that the bidscoiner converted the DICOMs into Nifti format.

Run BIDS Validator using the command below:

```
!deno run -ERWN jsr:@bids/validator $PROJECT_DIR/rawdata
```

If Deno is not installed, install Deno, then verify the installation by checking its version. use the command below to install and verify deno version.

```
import subprocess

# Download and install Deno
subprocess.run(["bash", "-c", "curl -fsSL
https://deno.land/install.sh | sh"], check=True)

# The installer will print instructions to add Deno to PATH. We
need to capture these and apply them.
# For Colab, typically Deno is installed to /root/.deno/bin. Add
this to PATH.
os.environ["PATH"] = f"{os.environ['HOME']}/.deno/bin:
{os.environ['PATH']}"

# Verify Deno installation and version
result = subprocess.run(["deno", "--version"],
capture_output=True, text=True, check=True)
print(result.stdout)
```



Rerun the BIDS Validator after installing Deno.

The output must show no errors; warnings related to non-standard metadata may be reviewed but should not block the pipeline.

AUTOMATED QUALITY ASSESSMENT VIA MRIQC

- 3 Before intensive processing, MRIQC computes image quality metrics (IQMs) to identify problematic datasets³.

- 3.1 Execute MRIQC at the participant level to extract metrics for T1w and DWI data.

Command

Computes objective Image Quality Metrics (IQMs) and generates individual and group HTML reports for visual screening.

mriqc

```
await lmod.load(mriqc/24.0.2)
```

```
!mriqc $BIDS_DIR $DERIVATIVES/mriqc participant --participant-label  
<SUBJECT_ID> -w /content/workdir --mem_gb [MEMORY_GIGABYTE] --nprocs  
[N_CPUS] --no-sub
```

Checkpoint: Review the .html reports in the output directory. Pay specific attention to:

- **CJV (Coefficient of Joint Variation):** High values in T1w may indicate poor tissue contrast, affecting segmentation.
- **SNR (Signal-to-Noise Ratio):** Essential for 1.5T data; ensure SNR is sufficient for microstructural modeling.
- **Mean FD (Framewise Displacement):** Review Mean FD and exclude subjects exceeding study-specific motion thresholds (e.g., > **0.5mm** or **1.0mm** depending on population).

ANATOMICAL PREPROCESSING AND SURFACE GENERATION

- 4 fMRIPrep provides the structural scaffolding⁴. In AD research, accurate surface reconstruction is vital for characterizing cortical thinning and for subsequent multimodal alignment.

- 4.1 Run fMRIPrep with surface processing (*recon-all*) enabled.

Command

Performs comprehensive anatomical preprocessing, including brain extraction, and surface reconstruction

fMRIPrep

```
await lmod.load(fmrip/25.1.3)

# Ensure FreeSurfer license is present
!export FS_LICENSE=[PATH_TO_YOUR_FREESURFER_LICENSE]

!fmrip $BIDS_DIR $DERIVATIVES/fmrip participant --participant-label <SUBJECT_ID> --w /content/work_dir --mem [MEMORY_MEGABYTE] --nprocs [N_CPUS] --anat-only --fs-license-file $FS_LICENSE --fs-no-reconall
```

Checkpoint: Verify the *anat/* subfolder contains the preprocessed T1w scan and its corresponding brain mask. Ensure the *freesurfer/sub-XXX/surf* folder contains the necessary *.pial* and *.white* meshes.

WHITE MATTER HYPERINTENSITY SEGMENTATION VIA TRUENET

- 5 Small vessel disease often co-occurs with AD. Segmenting White Matter Hyperintensities (WMH) using FSL TRUENET allows for the quantification of vascular burden and the subsequent "masking" of lesions to improve registration^{5,6}.

Installation

TRUENET is an optional component of FSL 6.0.7.13 and newer. Use this command to Install TRUENET by enabling the `--extra truenet` flag.

Command

This command installs truenet

Install Truenet

```
!curl -Ls  
https://fsl.fmrib.ox.ac.uk/fsldownloads/fslconda/releases/getfsl.sh |  
sh -s -- ~/fsl/ --extra truenet
```

Or:

If you already have FSL 6.0.7.13 or newer installed, you can install TRUENET with the *update_fsl_release* command

Command

This command installs truenet into an existing fsl installation

update_fsl_release

```
!update_fsl_release --extra truenet
```

If you have a CUDA-capable GPU on your system, these commands will install a GPU-accelerated version of TRUENET. If you do not have a GPU available, a CPU version of TRUENET will be installed.

If you don't have a GPU available, but wish to install the GPU version, you can use the `--cuda` option, passing the desired CUDA version.

For example, if you wish to install FSL with a version of TRUENET compatible with CUDA 11.2:

```
!curl -Ls  
https://fsl.fmrib.ox.ac.uk/fsldownloads/fslconda/releases/getfsl.s  
h | sh -s -- ~/fsl/ --extra truenet --cuda 11.2
```

Or to install TRUENET into an existing FSL installation:

```
!update_fsl_release --extra truenet --cuda 11.2
```

After installation, make sure FSL and TrUE-Net are on your PATH. Run

```
!export FSLDIR=~/.fsl
!source $FSLDIR/etc/fslconf/fsl.sh
!export PATH=$FSLDIR/bin:$PATH
```

Checkpoint: Run

```
truenet --help

#and

prepare_truenet_data --help
```

to confirm installation.

5.1 Preprocessing and preparing data for truenet

A series of preprocessing operations needs to be applied to any image that you want to use truenet on (most commonly T1-weighted and/or FLAIR images)⁵.

The *prepare_truenet_data* command was executed within a subject-level loop to provide automated, high-throughput preprocessing for the complete dataset.

Command

This script iterates through all subject folders, identifies the appropriate FLAIR and T1w files and outputs the prepared data

prepare_truenet_data

```
%%bash
RAW_DIR=$PROJECT_DIR/rawdata
TRUNET_DIR=$DERIVATIVES/truenet

for SUB in ${RAW_DIR}/sub-*; do
    SUB_ID=$(basename ${SUB})

    echo -----
    echo Preparing TrUE-Net data for: ${SUB_ID}
    echo -----

    # Create a subject-specific output directory in truenet
    derivatives
    mkdir -p ${TRUNET_DERIV}/${SUB_ID}/prepared_data

    prepare_truenet_data --
    FLAIR=${SUB_PATH}/anat/${SUB_ID}_*FLAIR.nii.gz --
    T1=${SUB_PATH}/anat/${SUB_ID}_*T1w.nii.gz --
    outname=${TRUNET_DERIV}/${SUB_ID}/prepared/${SUB_ID}

done
```

This command expects to be given your unprocessed T1 and/or FLAIR images. It will then perform the following steps:

- reorients images to the standard MNI space
- performs skull-stripping of T1 and FLAIR
- performs bias field correction of T1 and FLAIR
- registers the T1-weighted image to the FLAIR using linear rigid-body registration
- creates a mask from a dilated and inverted cortical CSF tissue segmentation (combined with other deep grey exclusion masks, using FSL FAST) and the *make_bianca_mask* command in FSL BIANCA⁷.

- using the above mask, calculates a distance map from the ventricles and a distance map from the gray matter.

5.2 White matter hyperintensity segmentation is performed by applying the *truenet evaluate* command to the output of *prepare_truenet_data*.

This pipeline was validated using *mwsc* (Multi-way Spatial Consensus) pretrained model (Two channels, FLAIR and T1).

mwsc models are ideal for small datasets (less than 20 subjects) while *ukbb* models are better for larger ones. Your images need to also match relatively well to those used to train the model; alternatively, you can take a pretrained model and *fine tune* this on your data⁵.

A bash loop was utilized to segment all subjects in a single execution.

Command

Applies a pretrained deep learning model (*mwsc*) to segment White Matter Hyperintensities (WMH).

truenet evaluate

```
%%bash
for SUB in ${TRUNET_DIR}/sub-*; do
  SUB_ID=$(basename ${SUB})

  echo Segmenting WMH for: ${SUB_ID}

  mkdir -p ${TRUNET_DIR}/${SUB_ID}/truenet_results

  truenet evaluate -m mwsc -i
  ${TRUNET_DIR}/${SUB_ID}/prepared_data -o
  ${TRUNET_DIR}/${SUB_ID}/truenet_results
done
```

Outputs:

- *Predicted_output_truenet_bin.nii.gz* (binary WMH mask)
- *Predicted_output_truenet_prob.nii.gz* (probability map)

Checkpoint: Visually inspect the *Predicted_output_truenet_bin.nii.gz* overlaid on the FLAIR image to ensure accurate lesion detection.

DIFFUSION PREPROCESSING VIA QSIPREP

- 6 QSIPrep is chosen for its focus on microstructural reproducibility. It handles the complex "denoising → unwarping → motion correction" chain in a single, high-fidelity step⁸.
- 6.1 Execute QSIPrep with MP-PCA denoising enabled.

Command

A specialized BIDS-App that performs denoising, susceptibility distortion correction, and eddy-current/motion correction.

qsiprep

```
ml qsiprep/1.0.1
!qsiprep $BIDS_DIR $DERIVATIVES/qsiprep participant --participant-label <SUBJECT_ID> -w $QSIPREP_WORK_DIR --dwi-only --fs-license-file $FS_LICENSE --output-resolution [VOXEL_SIZE_IN_MM] --nprocs [N_CPUS] -mem [MEMORY_MEGABYTE]
```

Why QSIPrep?

- **Reproduction:** It utilizes a "gold-standard" workflow including FSL *eddy* for motion and eddy current correction and MRtrix3 *dwidenoise* for noise suppression.
- **Coregistration:** It automatically aligns the diffusion data to the fMRIPrep-generated T1w reference using Boundary-Based Registration (BBR).

Checkpoint: Inspect the QSIPrep visual report. Check the "DWI to T1w Coregistration" section for any misalignment. Verify that the susceptibility distortion correction has significantly reduced the warping typical of 1.5T echo-planar imaging.

MICROSTRUCTURAL RECONSTRUCTION AND MODELLING

- 7 This stage performs the numerical reconstruction of the diffusion signal.
- 7.1 In this protocol, we utilize a **Single-Shell 3-Tissue (SS3T)** estimation, which is highly effective for resolving crossing fibers in white matter using single-shell data⁹.

Command

Estimates white matter Fiber Orientation Distributions

qsirecon

```
await lmod.load(qsirecon/1.1.0)
```

```
!qsirecon $DERIVATIVES/qsiprep $DERIVATIVES/qsirecon participant --  
participant-label <SUBJECT_ID> --recon-spec [RECON_SPEC] -w  
$QSIRECON_WORK_DIR --fs-license-file $FS_LICENSE --fs-subjects-dir  
$FS_DIR --atlases 4S --output-resolution [VOXEL_SIZE_IN_MM] --nprocs  
[N_CPUS] --mem [MEMORY_MEGABYTE]
```

Reconstruction Specs

Note

The *mrtrix_singleshell_ss3t_ACT-hsvs* spec used here was selected for its robustness in characterizing fiber orientations (FODs) and structural connectivity for single shell data.

However, users should note:

- This specific spec generates connectivity matrices and FOD maps.
- If your research requires standard DTI metrics (FA, MD) or Free-Water Elimination (FWE) maps, you may replace *\$RECON_SPEC* with alternatives such as *dipy_dti* or *amico_fwe*.
- Check the [QSIRecon Documentation](#) for the list of validated workflows that best fit your data acquisition (single-shell vs. multi-shell).

Interpretation of Potential Microstructural Metrics

Note

Depending on the chosen --recon-spec, the following biomarkers may be extracted:

	A	B
	Metric	Biological Significance
	FOD/Connectivity	Derived from the default SS3T spec; represents white matter pathways and network density.
	FA/MD	General axonal integrity and water mobility; requires a tensor-based spec (e.g., dipy_dti).
	FWE-FA/MD	Tissue-specific indices corrected for extracellular fluid/edema; requires FWE-specific specs.

GLYMPHATIC ASSESSMENT: DTI-ALPS INDEX CALCULATION

- 8 This section details the implementation of the glymphatic clearance assessment using the Automated DTI-ALPS pipeline developed by Winniework^{10,11,12}. This pipeline provides a non-invasive proxy for glymphatic efficiency by calculating the ALPS index from the diffusivity in the direction of the perivascular spaces¹³.

8.1 Environment and Data Migration

Before running the scripts, the BIDS-formatted raw data must be migrated to the dedicated processing folder. This is done to prevent the output of the script from being placed in the \$BIDS_DIR, as the script is designed to place its output in the directory it is currently running in. Rename the files to match the script's expected input naming scheme.

You need to also copy the ICBM.nii.gz template from the Winniework cloned repo into each subject's directory under the \$DERIVATIVES/dti_alps directory. This template is the standardized template for the MNI space which is required by the first script to run linear registration and vector reorientation.

Clone the Pipeline Repository

First, clone the necessary scripts and templates from the source repository.



Command

Cloning of the repository

```
%cd $DERIVATIVES/dti_alps
```

```
!git clone https://github.com/Winniework/Automated-DTI-ALPS-  
pipeline.git
```

Data Migration and Organization

You can manually copy each subject's raw DWI data from the \$BIDS_DIR and the ICBM_FA.nii.gz template from the cloned repo into the \$DERIVATIVES/dti_alps directory or use the loop below to do that for all subjects.

Command

new command name

```
# Loop through subjects to organize data
for SUB_DIR in ${BIDS_DIR}/sub-*; do
  SUB=$(basename ${SUB_DIR})
  TARGET_DIR=${DERIVATIVES}/dti_alps/${SUB}

  echo Preparing ${SUB}...
  mkdir -p ${TARGET_DIR}

  # Copy the template from the cloned repo into each subject folder
  cp ${DERIVATIVES}/dti_alps/Automated-DTI-ALPS-
pipeline/ICBM_FA.nii.gz ${TARGET_DIR}/

  # Copy raw BIDS files (using wildcards to stay generic)
  cp ${SUB_DIR}/dwi/${SUB}_*dwi.nii.gz ${TARGET_DIR}/dti.nii.gz
  cp ${SUB_DIR}/dwi/${SUB}_*dwi.bval ${TARGET_DIR}/dti.bval
  cp ${SUB_DIR}/dwi/${SUB}_*dwi.bvec ${TARGET_DIR}/dti.bvec
done
```

8.2 Preprocessing and Tensor Reorientation

Before execution, the following containerized modules must be initialized to provide the necessary tools (MRtrix3, FSL, ANTs, and AFNI):

Command

new command name

```
# Load required software modules
await lmod.load(mrtrix3/3.0.4)
await lmod.load(fsl/6.0.7.18)
await lmod.load(ants/2.6.0)
await lmod.load(afni/25.2.03)
```

Workflow Description:

The first script, which is a shell script, performs the following chronological operations:

1. **Denoising and Artifact Correction:** Uses MRtrix3 to suppress thermal noise and remove Gibbs ringing artifacts.
2. **Pre-conditioning:** Executes eddy current correction and B1 field bias correction to ensure signal homogeneity.
3. **Tensor Fitting:** Runs FSL dtifit to generate the diffusion tensor and FA maps.
4. **Spatial Normalization:** Normalizes the FA map to the ICBM_FA template and uses vecreg to reorient the diffusion tensors into MNI space.
5. **Diffusivity Extraction:** Extracts the reoriented diffusivity components (Dxx, Dyy, Dzz) using AFNI.

The shell script is designed to run inside a single subject's folder. To run it for all subjects, we use a loop:

Command

new command name

```
%cd $DERIVATIVES/dti_alps
for SUB in sub-*; do
    if [ -d $SUB ]; then
        echo Processing $SUB...
        cd $SUB

        bash ../Automated-DTI-ALPS-pipeline/1_DTI_preprocessing.sh
        cd ..
    fi
done
```

Expected Outputs:

- dti_reoriented_Dxx.nii.gz, dti_reoriented_Dyy.nii.gz, dti_reoriented_Dzz.nii.gz
- DTI_reoriented_ColorMap.nii.gz

8.3 Automated ALPS Index Calculation

Workflow Description:

The second script is python script and it automates the final quantification. Unlike the shell script, this specific Python script is designed to loop through all sub-directories starting with "sub" from the location where it is executed. It identifies the lateral ventricle level, places ROIs in the projection and association fibers, and calculates the ratio of diffusivity.

Command

new command name

```
# Ensure you are in the dti_alps directory where all sub- folders  
exist  
%cd $DERIVATIVES/dti_alps  
  
# Execute the automated calculation  
!python Automated-DTI-ALPS-pipeline/2_Automated_DTI-ALPS.py
```

Checkpoint:

1. **ALPS Output:** Verify that results.csv contains valid numerical entries for all the subjects.
2. **Visual Validation:** Inspect the generated .jpg files (e.g., sub-01_l.jpg). Ensure the white circles (ROIs) are accurately centered within the projection and association fiber bundles on the color-coded FA map.
3. **Glymphatic Interpretation:** Higher ALPS indices generally indicate more efficient perivascular clearance, while lower indices may suggest glymphatic impairment often seen in the AD continuum¹⁴.

Protocol references

1. Gorgolewski, K. J., et al. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, 3, 160044.
2. Zwiers, M. P., Moia, S., & Oostenveld, R. (2022). BIDScoin: A User-Friendly Application to Convert Source Data to Brain Imaging Data Structure. *Frontiers in Neuroinformatics*, 15, 770608.
3. Esteban, O., et al. (2017). MRIQC: Advancing the automatic prediction of image quality in MRI from unseen sites. *PLOS ONE*, 12(9), e0184661.
4. Esteban, O., et al. (2019). fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*, 16(1), 111–116.
5. Sundaresan, V., Zamboni, G., Dinsdale, N. K., Rothwell, P. M., Griffanti, L., & Jenkinson, M. (2021). Comparison of domain adaptation techniques for white matter hyperintensity segmentation in brain MR images. *Medical Image Analysis*, 74, 102215. <https://doi.org/10.1016/j.media.2021.102215>
6. Strain, J. F., Rahmani, M., Dierker, D., Owen, C., Jafri, H., Vlassenko, A. G., Womack, K., Fripp, J., Tosun, D., Benzinger, T. L. S., Weiner, M., Masters, C., Lee, J.-M., Morris, J. C., & Goyal, M. S. (2024). Accuracy of TrUE-Net in comparison to established white matter hyperintensity segmentation methods: An independent validation study. *NeuroImage*, 285, 120494. <https://doi.org/10.1016/j.neuroimage.2023.120494>
7. Griffanti, L., Zamboni, G., Khan, A., Li, L., Bonifacio, G., Sundaresan, V., Schulz, U., Kuker, W., Battaglini, M., Rothwell, P., & Jenkinson, M. (2016). BIANCA (Brain Intensity AbNormality Classification Algorithm): A new tool for automated segmentation of white matter hyperintensities. *NeuroImage*, 141, 191–205.
8. Cieslak, M., et al. (2021). QSIprep: an integrative platform for preprocessing and reconstructing diffusion MRI. *Nature Methods*, 18(7), 775–778.
9. Dhollander, T., & Connelly, A. (2016). A novel iterative approach to estimate three-tissue compartment fractions from multi-shell diffusion-weighted MRI. *ISMRM Annual Meeting*.
10. Winniework. (2023). Automated-DTI-ALPS-pipeline [Software]. GitHub. <https://github.com/Winniework/Automated-DTI-ALPS-pipeline>
11. Tatekawa, H., et al. (2023). Improved reproducibility of diffusion tensor image analysis along the perivascular space (DTI-ALPS) index. *Japanese Journal of Radiology*, 41(4), 393–400.
12. Zhang, W., et al. (2021). Glymphatic clearance function in patients with cerebral small vessel disease. *Neuroimage*, 238, 118257.
13. Taoka, T., et al. (2017). Diffusion tensor image analysis along the perivascular space (DTI-ALPS) for evaluating interstitial fluid diffusivity and glymphatic function. *Japanese Journal of Radiology*, 35(11), 658–663.
14. Taoka, T., et al. (2022). The Glymphatic System and the DTI-ALPS Index: A Review. *Magnetic Resonance in Medical Sciences*, 21(2), 268–273.