
DiffusionKit Documentation

Release 1.2.1

Brainnetome Center, CASIA

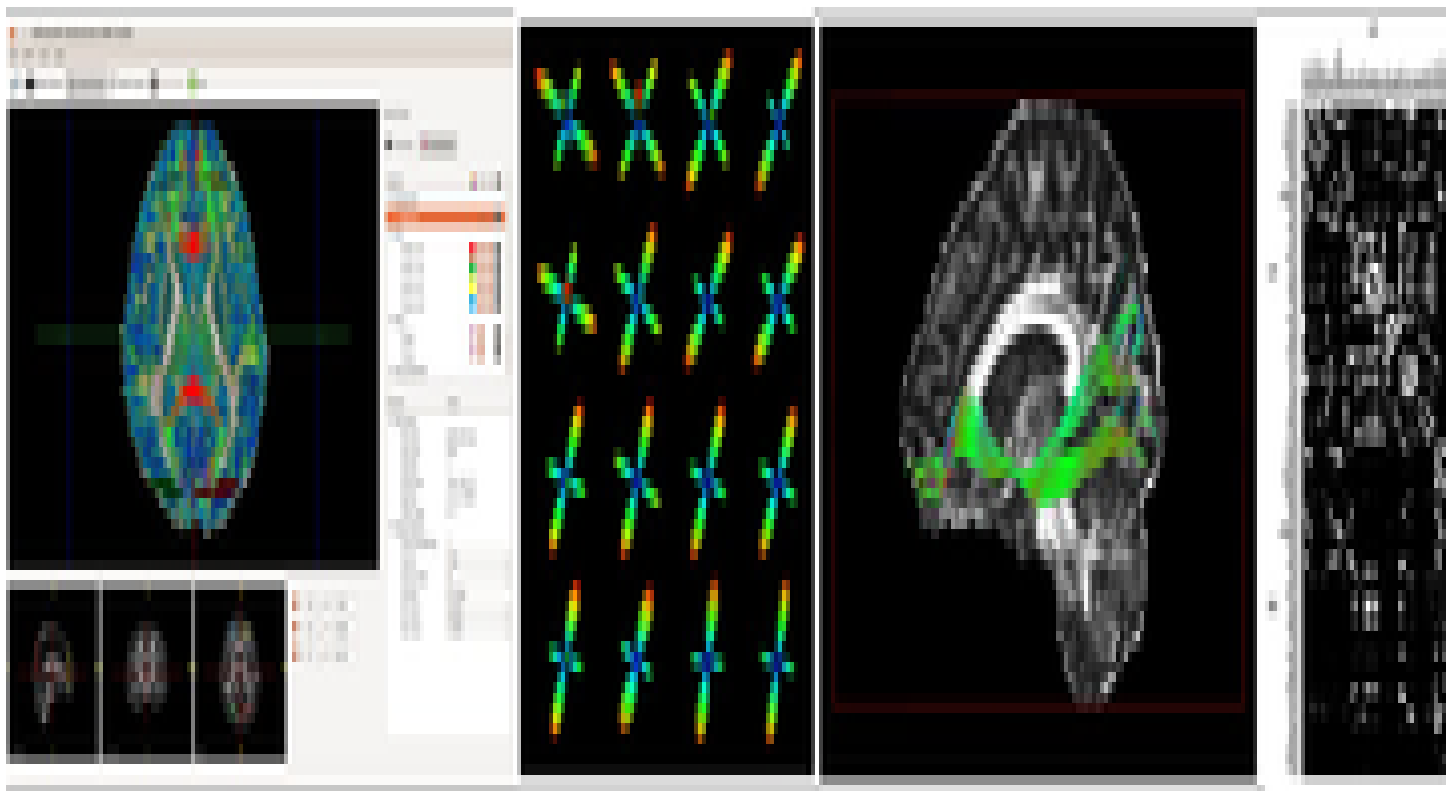
Apr 27, 2020

Contents

1	Welcome to Brainnetome DiffusionKit's Homepage	1
1.1	Manual	2
1.2	Tutorials	33
1.3	Screenshots	44
1.4	Download and Install	45
1.5	License	49
1.6	FAQs	50
1.7	Support	52

CHAPTER 1

Welcome to Brainnetome DiffusionKit's Homepage



Note:

- a full pipeline for (pre-)processing and visualization for diffusion MRI data.
- cross-platform support and a small installation size without 3rd party dependency.
- utilize a graphical interface along with command-line programs that enables easy operation and batch process-

ing.

Brainnetome DiffusionKit is a light one-stop cross-platform solution to dMRI data analysis. The package delivers a complete pipeline from data format conversion to preprocessing, from local reconstruction to fiber tracking, and from fiber statistics to visualization.

It was developed as a cross-platform framework, using VTK [2] for visualization, and Qt for GUI design. Both GPU and CPU computing were implemented for visualization to achieve high frame-rate, for rendering complex scene like whole brain tractographs in particular. The project was managed using the compiler-independent CMake [3], which is compatible with gcc/g++ and MS Visual Studio, etc. Well-established algorithms, such as the DICOM conversion tool dcm2nii by Chris Rorden [4] and the constrained spherical deconvolution (CSD) for HARDI reconstruction in MRtrix [5], were adopted with improved interface and user experience.

- Visit [Manual](#) page for a complete list of usage instructions.
- Visit [Tutorial](#) page to take a look at how DiffusionKit can solve your practical problems.
- Visit [Screenshot](#) page to see how the GUI front-end looks like.
- Visit [Support](#) page to submit comments, or email us at diffusion.kit@nlpr.ia.ac.cn.

Please see the navigation sidebar to the left to begin.

Important: DiffusionKit v1.5 Released! Please **Update** your DiffusionKit.

- Fix bugs.
 - Add features in 2D image display.
 - Add functions to transform fiber track format between DiffusionKit and MRtrix.
-

The citation for DiffusionKit could be:

Sangma Xie, Liangfu Chen, Nianming Zuo and Tianzi Jiang, **DiffusionKit: A Light One-Stop Solution for Diffusion MRI Data Analysis**, *Journal of Neuroscience Methods*, vol. 273, pp. 107-119, 2016. [\[PDF\]](#).

1.1 Manual

1.1.1 Overview

Brainnetome DiffusionKit is a light one-stop cross-platform solution to dMRI data analysis. The package delivers a complete pipeline from data format conversion to preprocessing, from local reconstruction to fiber tracking, and from fiber statistics to visualization. DiffusionKit was developed as a cross-platform framework, using ITK [1] for computation, VTK [2] for visualization, and Qt for GUI design. Both GPU and CPU computing were implemented for visualization to achieve high frame-rate, for rendering complex scene like whole brain tractographs in particular. The project was managed using the compiler-independent CMake [3], which is compatible with gcc/g++ and MS Visual Studio, etc. Well-established algorithms, such as the DICOM conversion tool dcm2nii by Chris Rorden [4] and the constrained spherical deconvolution (CSD) for HARDI reconstruction in MRtrix [5], were adopted with improved interface and user experience.

Key functions of the software

Func-tions	Program	Description (use '-h' argument for more details)
Preprocess	dcm2nii	Convert DICOM to unified 4D NIFTI files
	bneddy	Correct eddy current induced distortion and head motion
	topup/eddy	Correct eddy current and susceptibility induced distortion and head movements
	bet2	Extract brain tissue (Smith, 2002)
	bnsplit,bnmerge	Split/merge the 4D image along the 4th dimension
Modeling	bndti_estimate	Estimate tensor model, output FA, MD, tensor etc.
	bnhardi_ODF_estimate	Estimate ODF by SPFI method
	bnhardi_FOD_estimate	Estimate FOD by CSD method (Tournier et al., 2007)
Tracking	bndti_tracking	Track white matter fiber based on tensor model
	bnhardi_tracking	Track white matter fiber based on ODF/FOD
Visualize	bnviewer	Visualize various kinds of data (.nii.gz, .trk)
Tools	bncalc,bnroisplit	Numeric calculation and ROI generation
	bninfo	Show the head information of DICOM and NIFTI files
	reg_aladin,reg_f3d	Inter/intra-image registration across modalities
	reg_resample, reg_transform	Resample and apply transformation matrix
	bnfiber_end, bn- fiber_prune	Prune fiber bundle, logical and/or/not based on ROIs
	bnfiber_stats	Export attributes of fiber bundles
	bnfiber_map	Generate the fiber density going through each voxel
	bnnetwork	Function to construct anatomical network

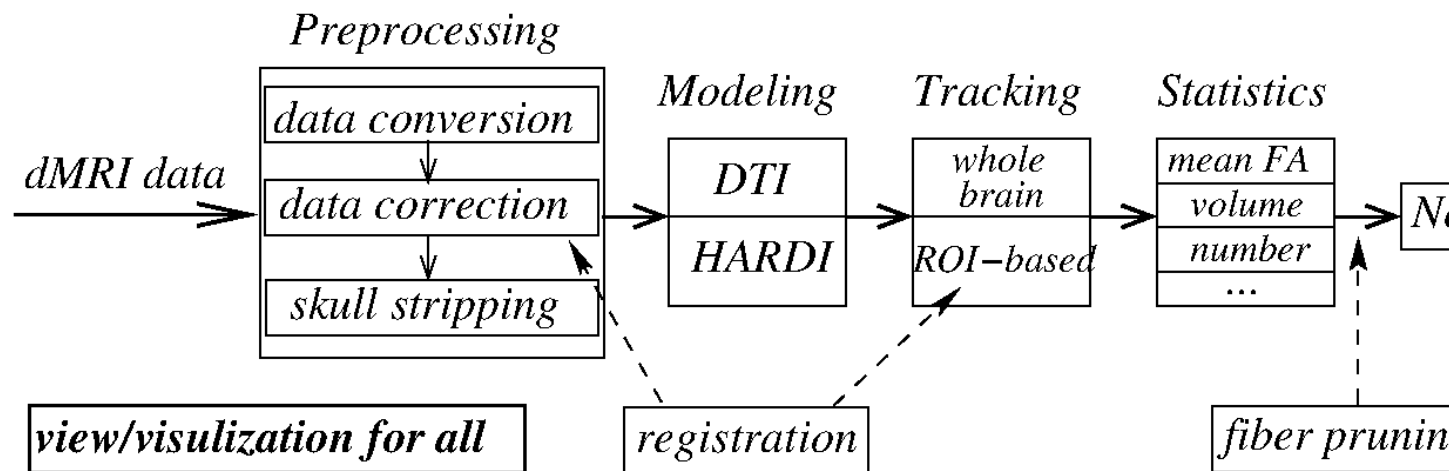


Fig. 1: Figure 1. The overall design framework of DiffusionKit

It should be noteworthy that, for all the computing steps provided in GUI, the called commands with the complete parameter list are displayed in a separate log window. Such a design is special for the users to keep in mind what he/she is doing, and furthermore, it could be directly copied into script (Bash, Python ...) for batch processing.

1.1.2 Data Processing Pipeline

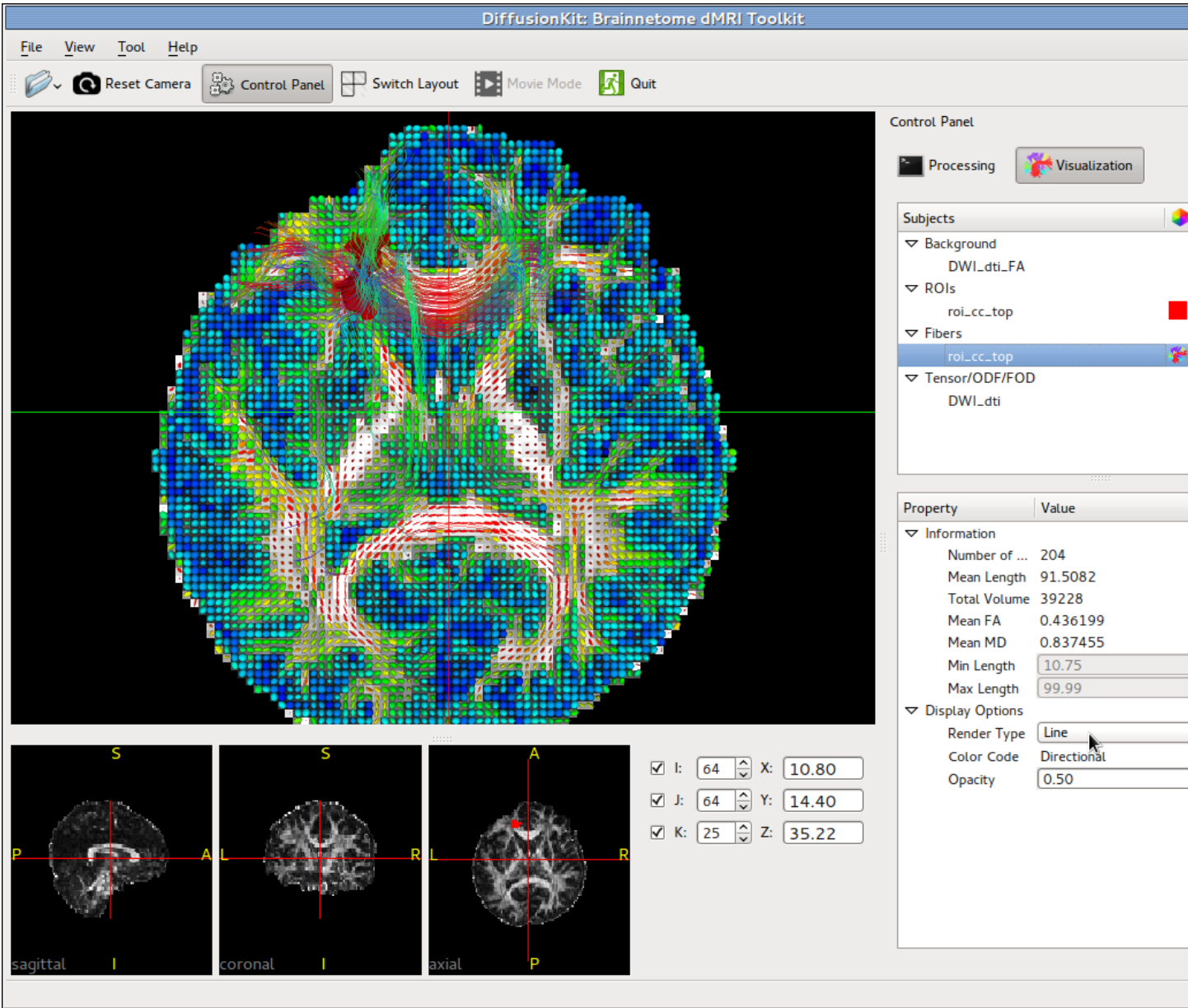


Fig. 2: Figure 2. The main window of the software.

Preprocessing

A set of command line tools for data preprocessing steps is described in this section, which include data format conversion, data correction and brain extraction.

DICOM to NIFTI Conversion

For saving space of data store, the nii.gz format was used throughout the whole pipeline. Firstly, we use the dcm2nii (by Chris Rorden, <https://www.nitrc.org/projects/dcm2nii>) to convert the data into a single 3/4D .nii.gz volume-series file, plus bval and bvec files. The format of bval file is

```
0 1500 1500 ... 1500
```

and the format of bvec file is

```
0 0.99944484233856 0.00215385644697 0.00269041745923 ...
0 -0.0053533311002 0.99836695194244 0.60518479347229 ...
0 0.03288444131613 -0.05708565562963 -0.79608047008514 ...
```

where the 0 in the first column indicates the b0 images in the scan and certainly the software also supports multi-b0 images. Since the DICOM formats from different scanner might be different, this function is not always able to successfully extract the bvec/bval files. If you encounter such a problem, please get back to us <link> and give the link of your data if it has big size beyond the email capability.

```
$ ./dcm2nii -h
Compression will be faster with /usr/local/bin/pigz
Chris Rorden's dcm2niiX version 24Nov2014
usage: dcm2nii [options] <in_folder>
Options :

-h    show help
-f    filename (%c=comments %f=folder name %p=protocol %i ID of
      patient %n=name of patient %s=series, %t=time; default 'DTI')
-o    output directory (omit to save to input folder)
-z    gz compress images (y/n, default n)

Defaults file : /home/ccm/.dcm2nii.ini
Examples :
dcm2nii /Users/chris/dir
dcm2nii -o /users/cr/outdir/ -z y ~/dicomdir
dcm2nii -f mystudy%s ~/dicomdir
dcm2nii -o "~/dir with spaces/dir" ~/dicomdir
Example output filename: '/DTI.nii.gz'
```

Data correction

During the MRI scanning, many factors can cause distortions and misalignment. The diffusion weighted imaging (DWI) volume series (as a 4D zipped NIFTI file) are then corrected for the distortions induced by off-resonance field and the misalignment caused by subject motion. The off-resonance effects are usually caused by the eddy currents of switching the diffusion encoding gradients and the susceptibility distribution of the imaged subjects, resulting in the deterioration of images due to blurring, spatial distortion, local signal artifacts, etc. The motion effects also cause image blurring and geometric misalignment [21] [22]. To correct the distortions induced by susceptibility and eddy current when the data is acquired with different phase-encode parameters, we include the correction mechanism using different phase-encode information. We exported the functions of topup, applytopup, eddy and eddy_combine

from FSL [21] [23] [24], compiled them on both Linux and Windows platforms, and packed the executable files into DiffusionKit. The detailed usage information, please refer to the website of FSL (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>).

```
$ ./topup -h
Part of FSL (build 504)
topup

Usage:
topup --imain=<some 4D image> --datain=<text file> --config=<text file with
↳parameters> --coutname=my_field

Compulsory arguments (You MUST set one or more of):
    --imain          name of 4D file with images
    --datain         name of text file with PE directions/times

Optional arguments (You may optionally specify one or more of):
    --out            base-name of output files (spline coefficients (Hz) and
↳movement parameters)
    --fout           name of image file with field (Hz)
    --iout           name of 4D image file with unwarped images
    --logout         Name of log-file
    --warpres        (approximate) resolution (in mm) of warp basis for the
↳different sub-sampling levels, default 10
    --subsamp        sub-sampling scheme, default 1
    --fwhm           FWHM (in mm) of gaussian smoothing kernel, default 8
    --config         Name of config file specifying command line arguments
    --miter          Max # of non-linear iterations, default 5
    --lambda         Weight of regularisation, default depending on --ssqlambda and
↳--regmod switches. See user documetation.
    --ssqlambda      If set (=1), lambda is weighted by current ssq, default 1
    --regmod         Model for regularisation of warp-field [membrane_energy
↳bending_energy], default bending_energy
    --estmov         Estimate movements if set, default 1 (true)
    --minmet         Minimisation method 0=Levenberg-Marquardt, 1=Scaled Conjugate
↳Gradient, default 0 (LM)
    --splineorder    Order of spline, 2->Qadratic spline, 3->Cubic spline. Default=3
    --numprec        Precision for representing Hessian, double or float. Default
↳double
    --interp         Image interpolation model, linear or spline. Default spline
    --scale          If set (=1), the images are individually scaled to a common
↳mean, default 0 (false)
    --regrid         If set (=1), the calculations are done in a different grid,
↳default 1 (true)
    -v,--verbose     Print diagnostic information while running
    -h,--help        display help info
```

```
$ ./eddy -h
Part of FSL (build 504)
eddy
Copyright(c) 2011, University of Oxford (Jesper Andersson)
```

```
Usage:
eddy --monsoon

Compulsory arguments (You MUST set one or more of):
    --imain File containing all the images to estimate distortions for
    --mask  Mask to indicate brain
    --index File containing indices for all volumes in --imain into --acqp and --
```

↳topup

(continues on next page)

(continued from previous page)

```
--acqp  File containing acquisition parameters
--bvecs File containing the b-vectors for all volumes in --imain
--bvals File containing the b-values for all volumes in --imain
--out   Basename for output
```

Optional arguments (You may optionally specify one or more of):

```
--session  File containing session indices for all volumes in --imain
--topup    Base name for output files from topup
--flm      First level EC model (linear/quadratic/cubic)
--fwhm     FWHM for conditioning filter when estimating the parameters
--niter    Number of iterations (default 5)
--resamp    Final resampling method (jac/lsr)
--repol    Detect and replace outlier slices
-v,--verbose  switch on diagnostic messages
-h,--help    display this message
```

Unfortunately, most clinical acquisitions do not currently meet the requirement (two or more acquisitions where the parameters are different so that the mapping fields for distortion correction are different.) of `topup`. To handle this issue, we implemented a function called `bneddy` to correct eddy-current induced distortion and head movements efficiently. `bneddy` applies rigid and affine registrations to amend the distortions and misalignment.

```
$ ./bneddy -h
bneddy: Eddy Currents and Head Motion Correction.
(Mar 10 2016, 16:20:52)

-i          Input file, 4D .nii.gz file.
-o          Prefix for output file, and it will also output the log file as
↳Prefix.txt which will be called by bnrotate_bvec.
-ref       0      Reference image.
-omp       2      Max number of threads
```

Skull Stripping (Brain Extraction)

This module is to strip the brain skull and extract the brain tissue, including gray matter, white matter, cerebrospinal fluid (CSF) and cerebellum. It largely benefits the following processing and analysis, offering better registration/alignment results and reducing computational time by excluding non-brain tissue. Therefore, although this step is not compulsory, we strongly recommend enforcing it. This module is adapted from the FSL/BET functions (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET>) for its excellent efficacy and efficiency.

```
$ ./bet2 -h
Part of FSL (build 504)
BET (Brain Extraction Tool) v2.1 - FMRIB Analysis Group, Oxford
Usage:
bet2 <input_filerooot> <output_filerooot> [options]
Optional arguments (You may optionally specify one or more of):
-o,--outline  generate brain surface outline overlaid onto original image
-m,--mask <m> generate binary brain mask
-s,--skull    generate approximate skull image
-n,--nooutput don't generate segmented brain image output
-f <f>        fractional intensity threshold (0->1); default=0.5; smaller
↳values give larger brain outline estimates
-g <g>        vertical gradient in fractional intensity threshold (-1->1);
↳default=0; positive values give larger brain outline at bottom, smaller at top
-r,--radius <r> head radius (mm not voxels); initial surface sphere is set to
↳half of this
```

(continues on next page)

(continued from previous page)

```

-w,--smooth <r> smoothness factor; default=1; values smaller than 1 produce
→more detailed brain surface, values larger than one produce smoother, less detailed
→surface
-c <x y z>      centre-of-gravity (voxels not mm) of initial mesh surface.
-t,--threshold -apply thresholding to segmented brain image and mask
-e,--mesh      generates brain surface as mesh in vtk format
-v,--verbose   switch on diagnostic messages
-h,--help      displays this help, then exits

```

Reconstruction of the diffusion model

The reconstruction for diffusion model within pixels is one of the key topics in diffusion MRI research and it is also one of key modules of the software. At the current stage, we have integrated three modeling methods: one is the traditional Gaussian model (commonly known as DTI, diffusion tensor imaging), and the other two are for high angular resolution diffusion imaging (HARDI). For more detailed information please refer to our review paper [11] and [13].

DTI Reconstruction

The classical diffusion gradient sequence used in dMRI is the pulsed gradient spin-echo (PGSE) sequence proposed by Stejskal and Tanner. This sequence has 90° and 180° gradient pulses with duration time δ and separation time Δ . To eliminate the dependence of spin density, we need at least two measurements of diffusion weighted imaging (DWI) signals, i.e. $S(b)$ with the diffusion weighting factor b in Eq. (1) introduced by Le Bihan etc, and $S(0)$ with $b = 0$ which is the baseline signal without any gradient.

In the b -value in Eq. (1), γ is the proton gyromagnetic ratio, $\|\mathbf{G}\| = \|\mathbf{G}\| \|\mathbf{u}\|$ is the diffusion sensitizing gradient pulse with norm $\|\mathbf{G}\|$ and direction \mathbf{u} . $\tau = \Delta - \frac{1}{3}\delta$ is normally used to describe the effective diffusion time. Using the PGSE sequence with $S(b)$, the diffusion weighted signal attenuation $E(b)$ is given by Stejskal-Tanner equation,

where D is known as the apparent diffusion coefficient (ADC) which reflects the property of surrounding tissues. Note that in general ADC D is also dependent on G in a complex way. However, free diffusion in DTI assumes D is only dependent on the direction of G , i.e. \mathbf{u} . The early works in dMRI reported that ADC D is dependent on gradient direction \mathbf{u} and used two or three DWI images in different directions to detect the properties of tissues. Then Basser et al. introduced diffusion tensor [12] to represent ADC as $D(\mathbf{u}) = \mathbf{u}^T \mathbf{D} \mathbf{u}$, where \mathbf{D} is called as the diffusion tensor, which is a 3×3 symmetric positive definite matrix independent of \mathbf{u} . This method is called as diffusion tensor imaging (DTI) and is the most common method nowadays in dMRI technique. In DTI, the signal $E(b)$ is represented as

The diffusion tensor D can be estimated from measured diffusion signal samples through a simple least square method or weighted least square method [12], or more complex methods that consider positive definite constraint or Rician noise. If single b -value is used, the optimal b -value for tensor estimation was reported to be in the range of $(0.7 \sim 1.5) \times 10^3 \text{ s/mm}^2$, and normally about twenty DWI images are used in DTI in clinical study. Some useful indices can be obtained from tensor D . The most important three indices are fractional anisotropy (FA), mean diffusivity (MD) and relative anisotropy (RA) defined as

where $\lambda_1, \lambda_2, \lambda_3$ are the three eigenvalues of D and λ_m is the mean eigenvalue. MD and FA have been used in many clinical applications. For example, MD is known to be useful in stroke study. For more detailed information please refer to our review paper [13].

```

$ ./bndti_estimate -h
bndti_estimate: Diffusion Tensors Estimation.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:15)

```

(continues on next page)

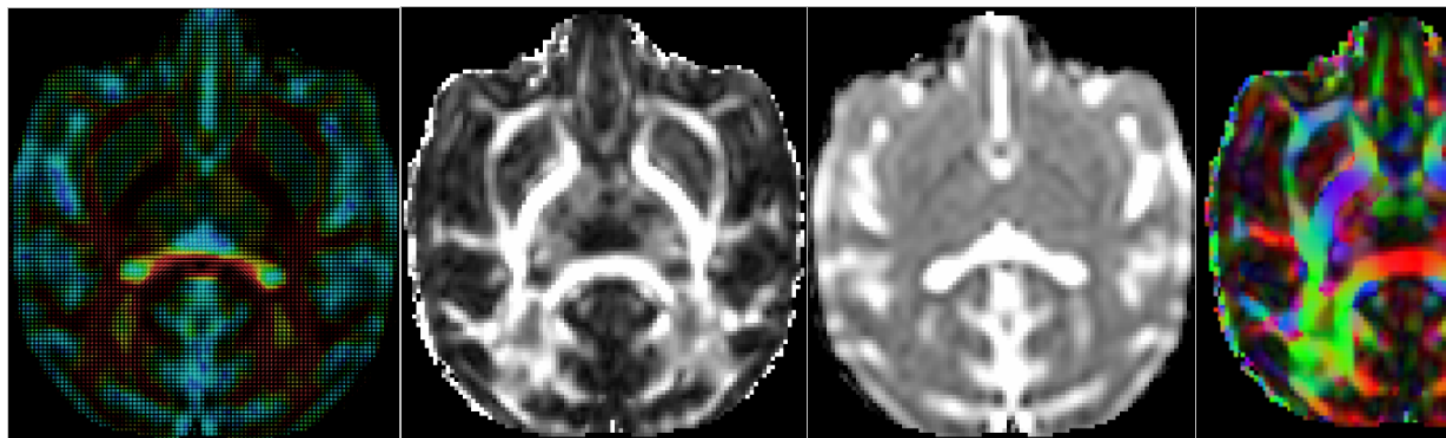


Fig. 3: Figure 4. Tensor field and the scalar maps estimated from a monkey data with $b = 1500\text{s/mm}^2$.

(continued from previous page)

```

general arguments
-d                Input DWI Data, in NIFTI/Analyze format (4D)
-g                Gradients direction file
-b                b value file
-m                Brain mask : filein mask | OPTIONAL
-o              dti  Result DTI : fileout prefix; 'dti' by default
-tensor         0    Save tensor : 0 - No; 1 - Yes; (Default: 0)
-eig            0    Save eigenvalues and eigenvectors : 0 - No; 1 - Yes;
↪ (Default: 0)

```

SPFI Reconstruction

It was proposed that the SPFI method has more powerful capability to identify the tangling fibers [8]. In SPFI [8], the diffusion signal is represented by spherical polar Fourier (SPF) basis functions in Eq. 7.

The SPF basis denoted by $R_{\{n\}}(|q|)Y_{\{l\}}^m(u)$ is a 3D orthonormal basis with spherical harmonics in the spherical portion and the Gaussian-Laguerre function in the radial portion. Furthermore, Cheng and his colleagues proposed a uniform analytical solution to transform the coefficients $a_{\{lm\}}$ of $E(q)$ to the coefficients $c_{\{lm\}}^{\Phi_w}$ of ODF (orientation distribution function) represented by the spherical harmonics basis, as in Eq. 8.

It is a model-free, regularized, fast and robust reconstruction method which can be performed with single-shell or multiple-shell HARDI data to estimate the ODF proposed by Wedeen et al. [14]. The implementation of analytical SPFI includes two independent steps. The first estimates the coefficients of $E(q)$ with least squares, and the second transforms the coefficients of $E(q)$ to the coefficients of ODF.

```

$ ./bnhardi_ODF_estimate -h
bnhardi_ODF_estimate: Orientation Distribution Function Estimation (SPFI method).
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:28)
-d                Input DWI data (4D NiftI file).
-b                Text file for b-values.
-g                Text file for grad directions.
-m                Input brain mask.
-o                Output prefix for EAP profile and ODF file.

```

(continues on next page)

(continued from previous page)

```

-scale          -1          The scale parameter of Spherical Polar Fourier Basis.
↳The default is calculated by the program.
-tau            0.02533     The diffusion time of DWI data.
-outGFA         false      Whether output generalized FA. True means Yes.
-rdis           0.015      The radius value for EAP profile.
-sh             4           Order of spherical basis
-ra             1           Order of radial basis
-lambda_sh      0           Regularization parameter for spherical basis
-lambda_ra      0           Regularization parameter for radial basis

```

CSD Reconstruction

The CSD method was proposed by Tournier et al. [9], which expresses the diffusion signal as in Eq. 9,

where $F(\theta, \Phi)$ is called the fiber orientation density function (fODF), which needs to be estimated, and $R(\theta)$ is the response function, which is the typical signal generated from one fiber. The response function can be directly estimated from diffusion weighted image (DWI) data by measuring the diffusion profile in the voxels with the highest fractional anisotropy values, indicating that a single coherently oriented fiber population is contained in these voxels. When the response function is obtained, we can utilize the deconvolution of $R(\theta)$ from $F(\theta, \Phi)$ to estimate the fiber ODF. The computation of the fiber ODF was carried out using the software MRtrix (J-D Tournier, Brain Research Institute, Melbourne, Australia, <http://www.brain.org.au/software/>). We thank Dr. Jacques-Donald Tournier for sharing MATLAB code of the CSD method, which inspired an efficient C/C++ implementation.

```

$ ./bnhardi_FOD_estimate -h
bnhardi_FOD_estimate: Constrained Spherical Deconvolution (CSD) based HARDI
↳reconstruction.
DiffusionKit (v1.2), http://diffusion.brainnetome.org/.
(Jul 15 2015, 11:50:20)
general arguments
-d              Input DWI Data, in NIFTI/Analyze format (4D)
-g              Gradients direction file
-b              b value file
-m              Brain mask : filein mask | OPTIONAL
-outFA         1          Whether to output the FA of DTI
-o              Result CSD Estimate (.nii.gz)
-lmax          8          6/8/10, Max order of the adopted harmonical base
-fa            [0.75,0.95] The FA threshold considered as single fiber
-erode         -1         The unit is voxel: Remove the garbage near the
↳boundary of FA image,
                    for better estimating response function
-nIter         50         Max iteration number before aborting
-lambda        1          The regularization weight for optimization
-tau           0.1        The threshold on the FOD amplitude used to identify
↳negative lobes
-hr            300        300/1000/5000. The later get more accurate estimation
↳while more
                    time consuming, so use the first one unless your
↳computer
                    is powerful !
bnhardi_FOD_estimate: Constrained Spherical Deconvolution (CSD) based HARDI
↳reconstruction.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:29)
general arguments
-d              Input DWI Data, in NIFTI/Analyze format (4D)

```

(continues on next page)

(continued from previous page)

```

-g                      Gradients direction file
-b                      b value file
-m                      Brain mask : filein mask | OPTIONAL
-outFA                  1      Whether to output the FA of DTI
-o                      Result CSD Estimate (.nii.gz)
-lmax                   8      6/8/10, Max order of the adopted harmonical base
-fa                     [0.75,0.95] The FA thesshold considered as single fiber
-erode                  -1     The unit is voxel: Remove the garbage near the
↪boundary of FA image, for better estimating response function
-nIter                  50     Max iteration number before aborting
-lambda                 1      The regularization weight for optimization
-tau                    0.1    The threshold on the FOD amplitude used to identify
↪negative lobes
-hr                     300    300/1000/5000. The later get more acurate estimation
↪while more time consuming, so use the first one unless your computer is powerful !
-omp                    2      Max number of threads

```

Fiber tracking and attributes extraction

Fiber tracking is a critical way to construct the anatomical connectivity matrix. For the tracking based on tensors from DTI, an intuitive way is to link the neighboring voxels following their main directions (e.g. V1 in the eigenvector of the DTI) given a set of some stop criteria, such as maximum bending angle of the curve and minimum FA value, which is to ensure the target voxel is indeed white matter microstructure. This is the so called deterministic streamline tractography [15], as illustrated in Figure 5.

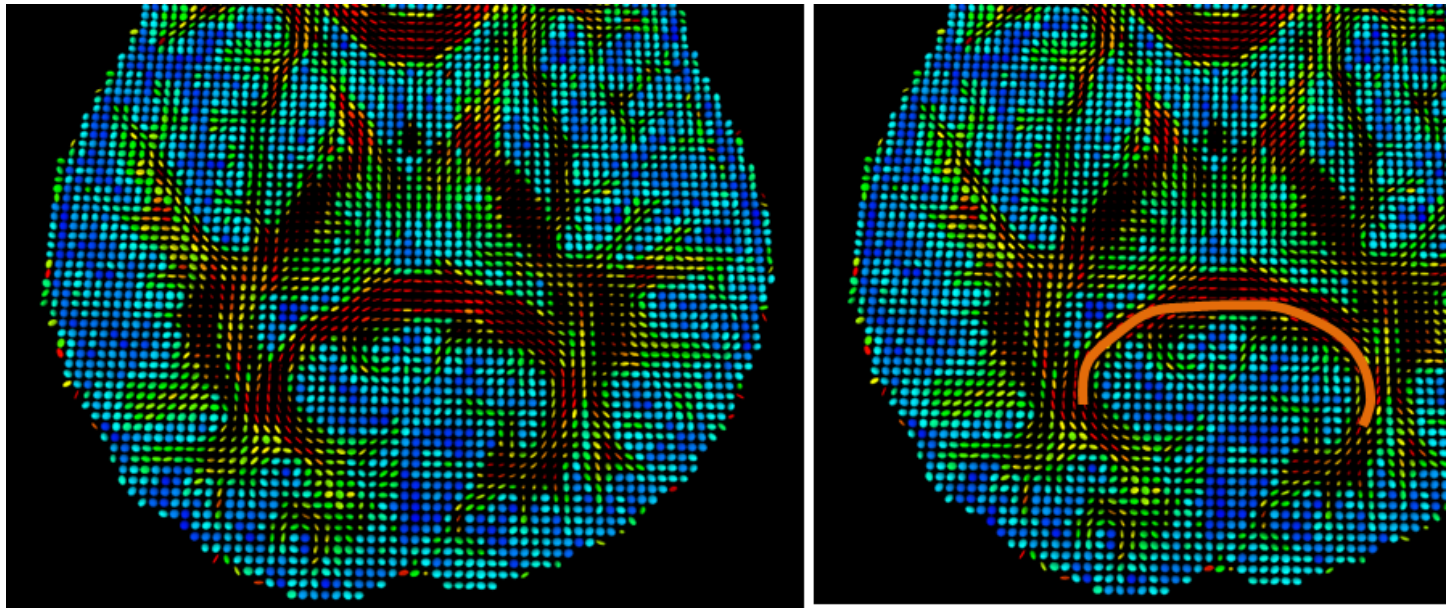


Fig. 4: Figure 5. Illustration for deterministic streamline tractography.

```

$ ./bndti_tracking -h
bndti_tracking: DTI Deterministic Fibertracking.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:29)
-d                      Input DTI data.

```

(continues on next page)

(continued from previous page)

```

-m          Mask Image.
-s          Seeds Image.
-fa         FA Image.
-invert     0      Invert x: 1; Invert y: 2; Invert z: 3.
-swap      0      Swap x/y: 1; Swap x/z: 2; Swap y/z: 3.
-ft        0.1    FA Threshold.
-at        45     Angular Threshold.
-sl        0.5    Step Length (Voxel).
-min       10     Threshold the fiber (mm). (remove fibers shorter than the_
↪number)
-max       5000   Upper-threshold the fiber (mm). (remove fibers longer than_
↪the number)
-o          Output Fibers Filename (.trk file).

```

The tracking module in the software for HARDI estimation is similar to the streamline method as described above. It should be kept in mind that, for HARDI estimation, usually there are more than one main direction (which is what we desired since it possibly identifies tangling fibers), so we should consider these kissing/branching cases. Meanwhile, since there is no explicit dominant directions for each voxel, searching algorithm should be applied to locate the main directions. The searching should run for each voxel, so it is not as fast as the traditional DTI tracking.

```

$ ./bnhardi_tracking -h
bnhardi_tracking: HARDI Deterministic Fibertracking.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:28)
-d          HARDI spherical harmonic image.
-fa         FA image.
-m          Mask image.
-s          Seeds image.
-sl        0.5    Step length (Voxel): float.
-ft        0.1    FA threshold: float.
-at        60     Angle threshold: float.
-min       10     Threshold the fiber (mm). (remove fibers shorter than_
↪the number)
-max       5000   Upper-threshold the fiber (mm). (remove fibers longer_
↪than the number)
-invert     0      Invert x: 1; Invert y: 2; Invert z: 3.
-swap      0      Swap x/y: 1; Swap x/z: 2; Swap y/z: 3.
-threshold  0.25  Threshold for the peaks of ODF.
-omp       1      Max number of threads.
-o          Tract file (.trk file).

```

Visualization for various images

The software provides a variety of entries for visualizing different data types, including 3D/4D image, Tensor/ODF/FOD image and white mater fibers. The views of different data could be superimposed for a precise anatomical localization. It should be noted that, if one uses the GUI, these view functions could be called in each processing steps for inspecting the results.

Figure 6. Illustrations for the capability of the software to show many types of images.

```

$ ./bnviewer -h
This program is the GUI frontend that displays and performs data reconstruction and_
↪fiber tracking on diffusion MR images, which have been developed by the teams of_
↪Brainnetome Center, CASIA.
basic usage:

```

(continues on next page)

(continued from previous page)

```

bnviewer [[-volume] DTI_FA.nii.gz]
         [-roi ROI/roi_cc_top.nii.gz]
         [-fiber ROI/roi_cc_top.trk]
         [-tensor DTI.nii.gz]/[-odf HARDI.nii.gz]

options:
  -help                show this help
  -volume .nii.gz      set input background data
  -roi .nii.gz         set input ROI data
  -fiber .trk          set input fiber data
  -tensor .nii.gz      set input DTI data (conflict with -odf args)
  -odf .nii.gz         set input ODF/FOD data (conflict with -tensor args)

```

Several other useful tools

Image registration

This module is implemented by NiftyReg, which is an open-source software for efficient medical image registration. It has been mainly developed by members of the Translational Imaging Group with the Centre for Medical Image Computing at University College London, UK [10]. In our software, the registration module is customized to auto-configure the parameter settings for different image modalities, e.g., two DWI images (for eddy current correction in current version), standard space and T1 space (for mapping ROIs to the individual space), DWI space and standard space (for statistical comparisons across subjects). This module contains several commands `reg_aladin` is the command for rigid and affine registration which is based on a block-matching approach and a Trimmed Least Square (TLS) scheme [18] [19]. `reg_f3d` is the command to perform non-linear registration which is based on the Free-From Deformation presented by Rueckert et al. [20]. `reg_resample` is been embedded in the package. It uses the output of `reg_aladin` and `reg_f3d` to apply transformation, generate deformation fields or Jacobian map images for example.

```

$ ./reg_aladin -h
Block Matching algorithm for global registration.
Based on Ourselin et al., "Reconstructing a 3D structure from serial histological
↪sections", Image and Vision Computing, 2001

Usage: reg_aladin -ref <filename> -flo <filename> [OPTIONS].
      -ref <filename> Reference image filename (also called Target or Fixed) ↪
↪(mandatory)
      -flo <filename> Floating image filename (also called Source or moving) ↪
↪(mandatory)

OPTIONS
      -noSym                The symmetric version of the algorithm is used by ↪
↪default. Use this flag to disable it.
      -rigOnly              To perform a rigid registration only. (Rigid+affine by ↪
↪default)
      -affDirect            Directly optimize 12 DoF affine. (Default is rigid ↪
↪initially then affine)
      -aff <filename>      Filename which contains the output affine ↪
↪transformation. [outputAffine.txt]
      -inaff <filename>    Filename which contains an input affine transformation.
↪ (Affine*Reference=Floating) [none]
      -rmask <filename>    Filename of a mask image in the reference space.
      -fmask <filename>    Filename of a mask image in the floating space. (Only ↪
↪used when symmetric turned on)

```

(continues on next page)

(continued from previous page)

```

    -res <filename>      Filename of the resampled image. [outputResult.nii]
    -maxit <int>         Maximal number of iterations of the trimmed least_
↪square approach to perform per level. [5]
    -ln <int>           Number of levels to use to generate the pyramids for_
↪the coarse-to-fine approach. [3]
    -lp <int>           Number of levels to use to run the registration once_
↪the pyramids have been created. [ln]
    -smooR <float>       Standard deviation in mm (voxel if negative) of the_
↪Gaussian kernel used to smooth the Reference image. [0]
    -smooF <float>       Standard deviation in mm (voxel if negative) of the_
↪Gaussian kernel used to smooth the Floating image. [0]
    -refLowThr <float>   Lower threshold value applied to the reference image._
↪[0]
    -refUpThr <float>    Upper threshold value applied to the reference image._
↪[0]
    -floLowThr <float>   Lower threshold value applied to the floating image._
↪[0]
    -floUpThr <float>    Upper threshold value applied to the floating image._
↪[0]
    -nac               Use the nifti header origin to initialise the_
↪transformation. (Image centres are used by default)
    -cog               Use the input masks centre of mass to initialise the_
↪transformation. (Image centres are used by default)
    -interp            Interpolation order to use internally to warp the_
↪floating image.
    -iso               Make floating and reference images isotropic if_
↪required.
    -pv <int>          Percentage of blocks to use in the optimisation scheme.
↪[50]
    -pi <int>          Percentage of blocks to consider as inlier in the_
↪optimisation scheme. [50]
    -speeeeed          Go faster
    -omp <int>         Number of thread to use with OpenMP. [4]
    -voff              Turns verbose off [on]

```

```

$ ./reg_f3d -h
Fast Free-Form Deformation algorithm for non-rigid registration.
Based on Modat et al., "Fast Free-Form Deformation using graphics processing units",_
↪CMPB, 2010

Usage: reg_f3d -ref <filename> -flo <filename> [OPTIONS].
    -ref <filename> Filename of the reference image (mandatory)
    -flo <filename> Filename of the floating image (mandatory)

OPTIONS
Initial transformation options (One option will be considered):
    -aff <filename>      Filename which contains an affine transformation_
↪(Affine*Reference=Floating)
    -incpp <filename>    Filename of floatf control point grid input

Output options:
    -cpp <filename>      Filename of control point grid [outputCPP.nii]
    -res <filename>      Filename of the resampled image [outputResult.nii]

Input image options:
    -rmask <filename>    Filename of a mask image in the reference space

```

(continues on next page)

(continued from previous page)

```

    -smooR <float>                Smooth the reference image using the specified_
↪sigma (mm) [0]
    -smooF <float>                Smooth the floating image using the specified_
↪sigma (mm) [0]
    --rLwTh <float>               Lower threshold to apply to the reference_
↪image intensities [none]. Identical value for every timepoint.*
    --rUpTh <float>               Upper threshold to apply to the reference_
↪image intensities [none]. Identical value for every timepoint.*
    --fLwTh <float>               Lower threshold to apply to the floating image_
↪intensities [none]. Identical value for every timepoint.*
    --fUpTh <float>               Upper threshold to apply to the floating image_
↪intensities [none]. Identical value for every timepoint.*
    -rLwTh <timepoint> <float>    Lower threshold to apply to the reference_
↪image intensities [none]*
    -rUpTh <timepoint> <float>    Upper threshold to apply to the reference_
↪image intensities [none]*
    -fLwTh <timepoint> <float>    Lower threshold to apply to the floating image_
↪intensities [none]*
    -fUpTh <timepoint> <float>    Upper threshold to apply to the floating image_
↪intensities [none]*

```

Spline options:

```

    -sx <float>                   Final grid spacing along the x axis in mm (in voxel if_
↪negative value) [5 voxels]
    -sy <float>                   Final grid spacing along the y axis in mm (in voxel if_
↪negative value) [sx value]
    -sz <float>                   Final grid spacing along the z axis in mm (in voxel if_
↪negative value) [sx value]

```

Regularisation options:

```

    -be <float>                   Weight of the bending energy penalty term [0.005]
    -le <float> <float>           Weights of linear elasticity penalty term [0.0 0.0]
    -l2 <float>                   Weights of L2 norm displacement penalty term [0.0]
    -jl <float>                   Weight of log of the Jacobian determinant penalty term_
↪[0.0]
    -noAppJL                      To not approximate the JL value only at the control_
↪point position

```

Measure of similarity options:

NMI with 64 bins is used expect **if** specified otherwise

```

    --nmi                        NMI. Used NMI even when one or several other measures_
↪are specified.
    --rbn <int>                  NMI. Number of bin to use for the reference image_
↪histogram. Identical value for every timepoint.
    --fbn <int>                  NMI. Number of bin to use for the floating image_
↪histogram. Identical value for every timepoint.
    -rbn <tp> <int>              NMI. Number of bin to use for the reference image_
↪histogram for the specified time point.
    -rbn <tp> <int>              NMI. Number of bin to use for the floating image_
↪histogram for the specified time point.
    --lncc <float>               LNCC. Standard deviation of the Gaussian kernel._
↪Identical value for every timepoint
    -lncc <tp> <float>           LNCC. Standard deviation of the Gaussian kernel for_
↪the specified timepoint
    --ssd                        SSD. Used for all time points
    -ssd <tp>                   SSD. Used for the specified timepoint
    --kld                        KLD. Used for all time points

```

(continues on next page)

(continued from previous page)

-kld <tp>	KLD. Used for the specified timepoint
-amc	To use the additive NMI for multichannel data.
↪ (bivariate NMI by default)	
Optimisation options:	
-maxit <int>	Maximal number of iteration per level [300]
-ln <int>	Number of level to perform [3]
-lp <int>	Only perform the first levels [ln]
-nopy	Do not use a pyramidal approach
-noConj	To not use the conjugate gradient optimisation but a
↪ simple gradient ascent	
-pert <int>	To add perturbation step(s) after each optimisation.
↪ scheme	
F3D2 options:	
-vel	Use a velocity field integration to generate the.
↪ deformation	
-fmask <filename>	Filename of a mask image in the floating space
OpenMP-related options:	
-omp <int>	Number of thread to use with OpenMP. [4]
Other options:	
-smoothGrad <float>	To smooth the metric derivative (in mm) [0]
-pad <float>	Padding value [nan]
-voff	To turn verbose off

```
$ ./reg_resample -h
Usage: reg_resample -ref <filename> -flo <filename> [OPTIONS].
      -ref <filename>      Filename of the reference image (mandatory)
      -flo <filename>      Filename of the floating image (mandatory)

OPTIONS
      -trans <filename>    Filename of the file containing the transformation.
↪ parametrisation (from reg_aladin, reg_f3d or reg_transform)
      -res <filename>      Filename of the resampled image [none]
      -blank <filename>    Filename of the resampled blank grid [none]
      -inter <int>         Interpolation order (0, 1, 3, 4) [3] (0=NN, 1=LIN, 3=CUB,
↪ 4=SINC)
      -pad <int>           Interpolation padding value [0]
      -tensor              The last six timepoints of the floating image are
↪ considered to be tensor order as XX, XY, YY, XZ, YZ, ZZ [off]
      -psf                 Perform the resampling in two steps to resample an image.
↪ to a lower resolution [off]
      -voff               Turns verbose off [on]
```

Need further help?

Check the website <http://cmictig.cs.ucl.ac.uk/wiki/index.php/NiftyReg>.

Image calculation and ROI generation

The module `bncalc` provides simple image calculations, such as add/minus/multiply/divide operations. Meanwhile, it can also generate user-defined ROIs given the origin and radius in a user-specified image space.

```

$ ./bncalc -h
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
This function provide basic process for the input data (NIFTI/Analyze format)
Usage of bncalc:
  -i      image      The original file you want to manage.
  -add    image/value Add to the data from the last step.
  -sub    image/value Subtract data from the last step.
  -mul    image/value Multiply the data from the last step.
  -div    image/value Divide the data from the last step.
  -roi    x,y,z,r     Generate a ROI centered at [x,y,z] (MMI mm) with radius r,
↳ in the input data space.
  -roi_rect x1,x2,y1,y2,z1,z2 OR i
                                Generate a cuboid ROI by specifying left-bottom
                                corner [x1,y1,z1] and right-up corner [x2,y2,z2].
                                If only one integer is specified, we assume you
                                want to get the i-the volume along the 4th dimension (from
↳ 0).
  -mask    image      Mask the data from last step by this input one.
                                If this input is a binary, then it is the same as
                                -mul, otherwise it keep the voxels from the last step
                                when the new input is nonzero.
  -bin     value      set 1 if >value, otherwise 0.
  -uthr    value      Set voxel=0 when it>value.
  -dthr    value      Set voxel=0 when it<value.
  -o       image      output a NIFTI (.nii.gz) file

```

Fiber manipulation

After obtaining a large bundle of white matter fibers, you may want to prune the fibers that go through specified locations (ROIs). Here, we provide several tools to manipulate the fiber bundles. `bnfiber_prune`, to split fiber bundles based on given ROIs.

```

$ ./bnfiber_prune
bnfiber_prune: Prune fiber bundles based on given ROIs.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:29)
  -fiber    Tract file (.trk file)
  -and      AND ROI file: roi1.nii.gz,roi2.nii.gz
  -or       OR ROI file: roi1.nii.gz,roi2.nii.gz
  -not      NOT ROI file: roi1.nii.gz,roi2.nii.gz
  -o        Output Tract file (.trk file).

```

`bnfiber_end`, to cut the fiber bundles given start/stop ROIs, which is useful to get the exact connections between two ROIs in constructing connectivity matrix.

```

$ ./bnfiber_end -h
bnfiber_end: Extract fibers which end in the two given rois.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:29)
  -fiber    Tract file (.trk file).
  -roi1     ROI1 file
  -roi2     ROI2 file
  -o        Output tract file (.trk file)

```

`bnfiber_stats`, to extract statistical properties of the fiber bundle, such as mean FA/MD and fiber numbers.

```
$ ./bnfiber_stats -h
bnfiber_stats: Show fiber stats.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:33)
-fiber          Input tract file (.trk file).
```

bnfiber_map, to compute the fiber density map which is used in track density imaging [16].

```
$ ./bnfiber_map -h
bnfiber_map: Calculate the fiber density.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
(Jun 24 2016, 09:40:29)
-fiber          Fiber file (.trk).
-o              Output file (NIfTI).
-nor            1      Normalize fiber density: 1 (yes) or 0 (no).
```

bnmerge / bnsplit, to merge the 3D volumes to 4D or split 4D to 3D volumes.

```
$ ./bnmerge -h
bnmerge: Merge 3D NIfTI files to 4D NIfTI file.
This program is to merge 3D NIfTI files to 4D NIfTI file.
Usage: bnmerge filein1 filein2 ... fileout
options:      -h          : show this help
```

```
$ ./bnsplit -h
bnsplit: Split 4D volume to 3D volumes.
This program splits 4D volume to 3D volumes.
basic usage: split -i FILE_IN -o FILE_OUT prefix
options:      -h          : show this help
              -v LEVEL   : the verbose level to LEVEL
```

bninfo, to display a short head information of the input image. Supported input image format includes NIFTI/DICOM.

```
$ ./bninfo -h
Show file header information.
DiffusionKit (v1.3), http://diffusion.brainnetome.org/.
Options:
-h,--help          Print this help information
-c,--is-canonical  Check XFORM and parse whether transform matrix is canonical
```

1.1.3 GUI Front-end of Data Processing Pipeline

Preprocessing

A ‘Confirmation’ option is provided in each of the following preprocessing step. When confirmation is required, the program displays the processing result after the target file is generated to ensure the correctness the final result.

DICOM to NIFTI Conversion

This is the GUI front end for dcm2nii program. Details can be found at *DICOM to NIFTI Conversion* section.

☒ **DICOM to NIFTI**

Input Data Dir: ...

Output Data File: ...

☐ Confirmation

Eddy Current Correction

This is the GUI front end for `bneddy` program. Details can be found at [Data correction](#) section.

☒ **Eddy Current Correction**

Input Data File: ...

Output Data File: ...

☐ Confirmation

Reference:

Since this is a time-consuming step in preprocessing stage, one can skip this step ignoring potential image registration error in original DWI data by unchecking the option. Then the converted NIFTI image would be input of brain extraction tool directly if the skull stripping option is enabled.

Skull Stripping

FSL's [BET \(Brain Extraction Tool\)](#) is integrated as part of the program, so that one can verify automatic skull stripping result by visualizing overlaying mask image upon original image. See [Volume Image Overlay](#) on how to generate overlaid background images.

☒ **Skull Stripping**

Input File: ...

Brain File: ...

Mask File: ...

☐ Confirmation

Threshold:

A threshold value that indicate 'fractional intensity' is exposed on the GUI, so that one can adjust the threshold to get smaller/larger brain outline estimates. Details on the implement can be found at [BET's UserGuide](#).

Diffusion Model Reconstruction

Starting from this section, we provide three different approaches for diffusion MR data processing:

This is the GUI front end for *Reconstruction of the diffusion model*, which generates required model for *Fiber tracking and attributes extraction* from DWI image and related b-values.

Click one of the options above to see guidelines on using the specific method for data processing.

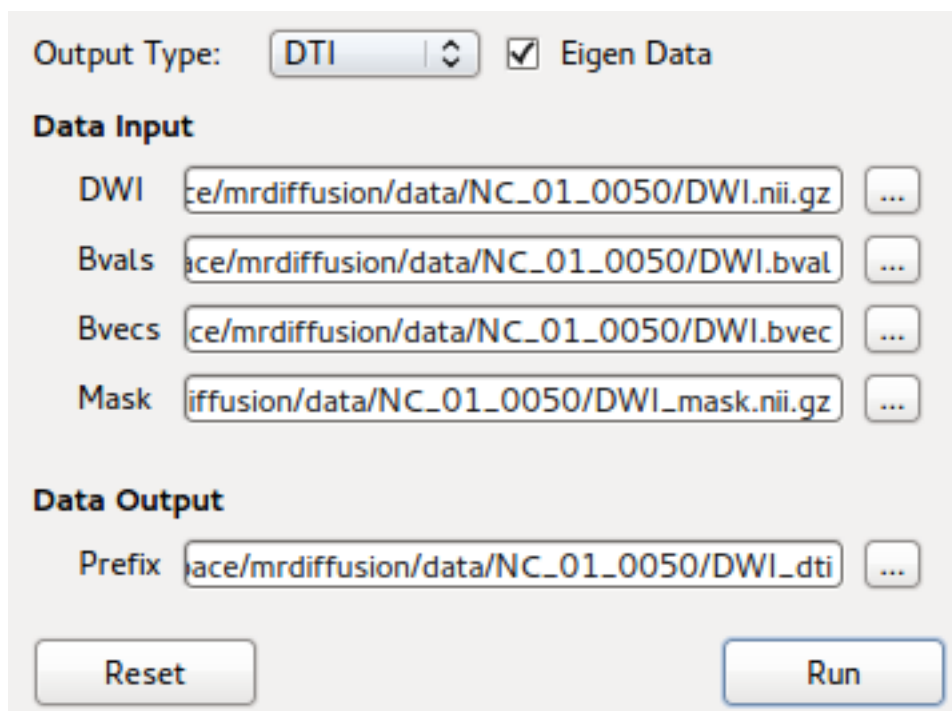


Fig. 5: GUI front-end for *DTI Reconstruction* using `bndti_estimate`

Once DWI data is selected through file selection dialog, all other input and output options would be automatically generated. This simplifies the usage of command-line program, which requires a list of options to be entered.

Fiber Tracking

To generate fiber tracking result using previously generated diffusion model, we provide the GUI front-end for using `bndti_tracking` and `bnhardi_tracking`, which generate tract files from DTI and HARDI respectively. The details of the program are described on *Fiber tracking and attributes extraction* section.

Fiber tracking require seed(ROI) files to initialize a tracking process. One can either generate the seed files from the **Registration Tool** we provided from **Tools** menu, or split a brain atlas image (e.g. AAL) using `bncalc` or `bnroisplit` into multiple ROI files. To generate whole brain fiber, just use the brain mask image generated during *Skull Stripping* as seed input. This may cost a few minutes when HARDI is used as input diffusion model.

Miscellaneous Tools

Image Registration

This is basically the GUI front-end of NiftyReg, while we perform `reg_aladin` on intra-subject images, and additionally run `reg_f3d` for inter-subject cases.

The details of the tool is described in *Image Registration* section in UserGuide page.

Output Type: HARDI ⬆ ⬆

Data Input

DWI ce/mrdiffusion/data/NC_01_0050/DWI.nii.gz ...

Bvals ce/mrdiffusion/data/NC_01_0050/DWI.bval ...

Bvecs ce/mrdiffusion/data/NC_01_0050/DWI.bvec ...

Mask iffusion/data/NC_01_0050/DWI_mask.nii.gz ...

Data Output

Prefix ce/mrdiffusion/data/NC_01_0050/DWI_spfi ...

Reconstrut Parameters

Method SPFI (ODF) ⬆ ⬆ ☐ Generate Direction

SH Order 4 ⬆ ⬆ SH Lambda 0.00 ⬆ ⬆

Rad Order 1 ⬆ ⬆ Rad Lambda 0.00 ⬆ ⬆

Amptitude 0.02533 ⬆ ⬆ Radius 0.015 ⬆ ⬆

☐ Generate GFA

Fig. 6: GUI front-end for *SPFI Reconstruction* using `bnhardi_ODF_estimate`

Output Type: HARDI | ⬆ ⬇ ⬆

Data Input

DWI ce/mrdiffusion/data/NC_01_0050/DWI.nii.gz ...

Bvals ce/mrdiffusion/data/NC_01_0050/DWI.bval ...

Bvecs ce/mrdiffusion/data/NC_01_0050/DWI.bvec ...

Mask iffusion/data/NC_01_0050/DWI_mask.nii.gz ...

Data Output

Prefix ce/mrdiffusion/data/NC_01_0050/DWI_fod ...

Reconstrut Parameters

Method CSD (FOD) | ⬆ ⬇ ⬆ ☐ Generate Direction

SH Order 8 ⬆ ⬇ ⬆ FA Range 0.75 ⬆ ⬇ ⬆

Max Iteration 50 ⬆ ⬇ ⬆ 0.95 ⬆ ⬇ ⬆

Regularization 1.00 ⬆ ⬇ ⬆ Resolution 300 ⬆ ⬇ ⬆

Amptitude 0.10 ⬆ ⬇ ⬆

Fig. 7: GUI front-end for *CSD Reconstruction* using `bnhardi_FOD_estimate`

Data Type HARDI

Data Input

Data /sion/data/NC_01_0050/DWI_spfi_ODF.nii.gz ...

Mask /diffusion/data/NC_01_0050/DWI_mask.nii.gz ...

FA /ffusion/data/NC_01_0050/DWI_spfi_FA.nii.gz ...

Seeds /diffusion/data/NC_01_0050/DWI_mask.nii.gz ...

Data Output

Fiber /mrdiffusion/data/NC_01_0050/DWI_mask.trk ...

Tracking Parameters

Angle threshold 45 FA Threshold 0.10

Step size (voxel) 0.5

Min length (mm) 10 Max length 5000

Image Registration

Options

☒ Intra-Subject ☐ Inter-Subject

Source /ome/liangfu/workspace/mrdiffusion/data/NC_01_0050/DWI_brain.nii.gz ...

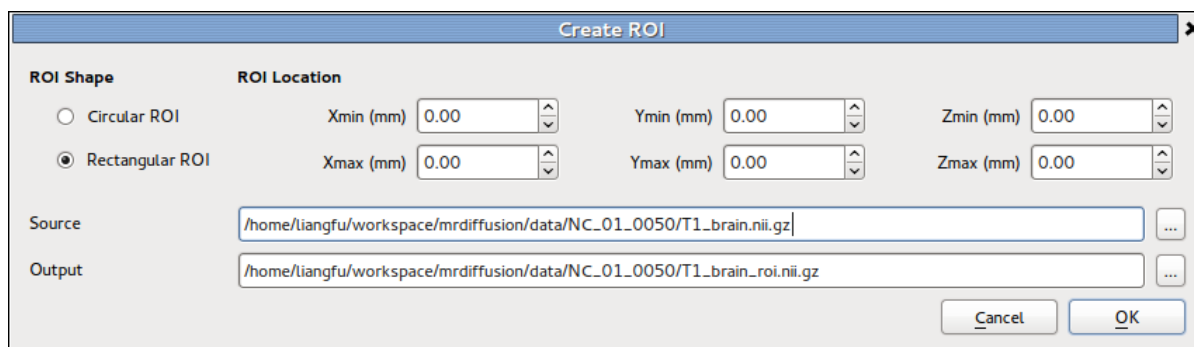
Output /ne/liangfu/workspace/mrdiffusion/data/NC_01_0050/DWI_brain_r.nii.gz ...

Reference /home/liangfu/workspace/mrdiffusion/data/NC_01_0050/T1_brain.nii.gz ...

Cancel OK

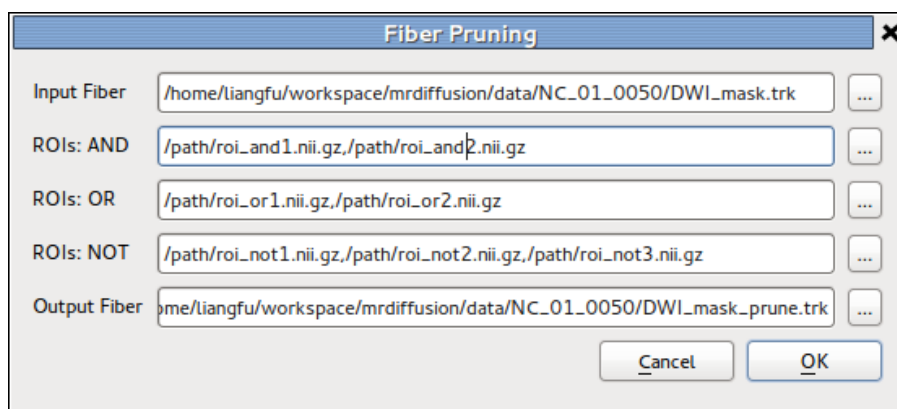
Create ROI

The details of the tool is described in [Image calculation and ROI generation](#) section in UserGuide page.



Fiber Pruning

The details of the tool is described in [Fiber manipulation](#) section in UserGuide page.



1.1.4 Guidelines on Data Visualization

In this page, the general functionality implemented in the `bnviewer` program is described in detail. This covers methods that a user can visualize almost all kinds of diffusion MRI data.

Except those specially noticed, all the data should be formatted in either NIFTI or ANAYLYZE image. NIFTI format is recommended, see [Data Format](#) page for more details.

Command Line Interface

Typing `bnviewer -h` in command line would given following information, which help users to load a list of files without click them one by one. The can be extremely useful when one wants to contantly load a large list of data files to get screen capture images.

```
basic uage:
  bnviewer [[-volume] DTI_FA.nii.gz]
           [-roi ROI/roi_cc_top.nii.gz]
```

(continues on next page)

(continued from previous page)

```

[-fiber ROI/roi_cc_top.fiber]
[-tensor DTI.nii.gz]/[-odf HARDI.nii.gz]
[-atlas DTI.nii.gz]

options:
  -help                show this help
  -volume .nii.gz      set input background data
  -roi .nii.gz         set input ROI data
  -fiber .fiber/.trk   set input fiber data
  -tensor .nii.gz      set input DTI data (conflict with -odf args)
  -odf .nii.gz         set input ODF/FOD data (conflict with -tensor args)

```

Todo: create interface for generating high quality images and save them into specified location.

Image Data Navigation

3D Navigation

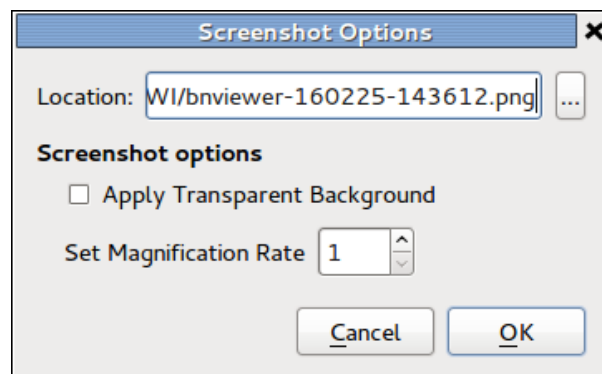
When an image data is loaded, one can change view angle by dragging *left* mouse button. This is the default interactor style implemented in *vtkInteractorStyleTrackballCamera*, which is used internally in our program. According to VTK's documentation:

vtkInteractorStyleTrackballCamera allows the user to interactively manipulate (rotate, pan, etc.) the camera, the viewpoint of the scene. In trackball interaction, the magnitude of the mouse motion is proportional to the camera motion associated with a particular mouse binding. For example, small left-button motions cause small changes in the rotation of the camera around its focal point. For a 3-button mouse, the left button is for *rotation*, the right button for *zooming*, the middle button for *panning* (translation), and ctrl + left button for *spinning*. (With fewer mouse buttons, ctrl + shift + left button is for zooming, and shift + left button is for panning.)

We simplify the trackball interaction to used middle button for both panning and zooming, in order to spare right mouse click to popup option menu for more actions.

The background color for 3D view is defined as black by default. This can be changed from context menu popped up when *right* mouse click event is captured. By selecting the color from the popup color dialog, the background color is changed instantly.

Also, saving screenshots as PNG image is enabled from the context menu. The default image is named after current timestamp to avoid naming duplication.



To customize generated image for different requirements, two additional options are provided, as shown in the above screenshot. The higher magnification value indicate higher resolution of generated image. By enabling transparent background in output image, one can alternate the background color in a later stage.

2D Navigation

Three 2D slice image views are placed below 3D view by default. The three slice views from left to right are *sagittal plane*, *coronal plane* and *axial plane* respectively.

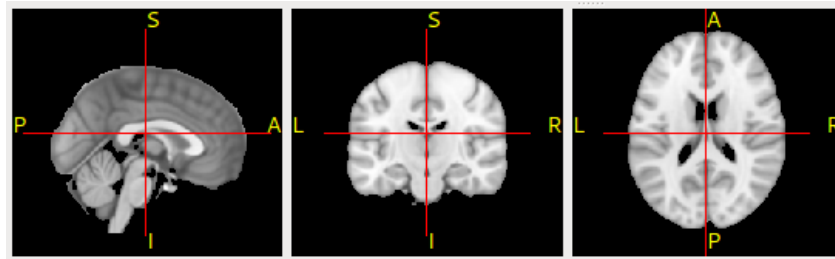


Fig. 8: *sagittal plane*, *coronal plane* and *axial plane* (left to right)

2D slice view widget sizes can be enlarged by switching the overall layout. This functionality is implemented in a single button, named **Switch Layout**, available on the toolbar. By toggling the button, the main view panels switches between 3D view port and a combination of sliced 2d views.

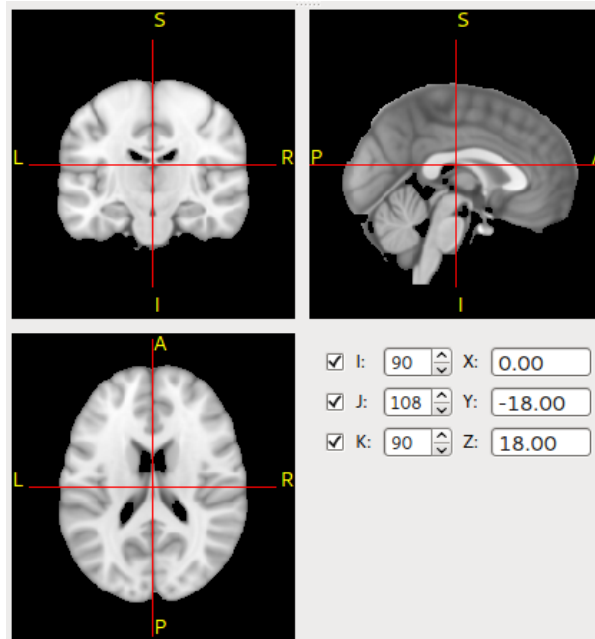


Image slice index can be changed by clicking on 2d image slice views. Besides, we perform radiological/neurological (or RAS/LAS) conversion instantly when slice index values are changed.

For people interested in more details about the radiological/neurological conversion, please refer to FSL's [Orientation Explained](#).

Background Volume Data Layer

Beginning from this section, several kinds of ‘layers’ are described. Each ‘layer’ may be contain one or more data that holds the same property. By default, the layers manager is located at top-right side of the window.

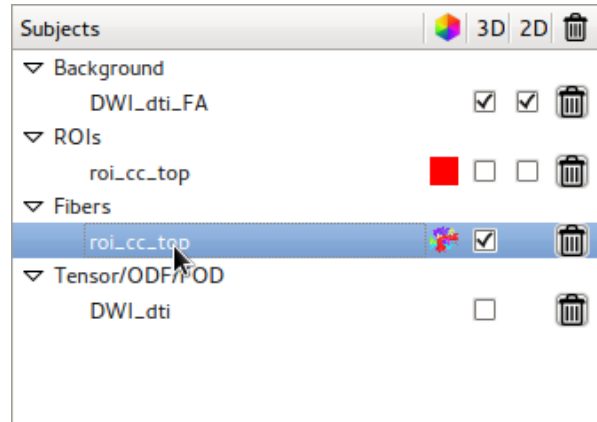


Fig. 9: The layers manager

We refer to *Background* data as plain 3D or 4D volume data that user wants to visualize its sliced views. The content should be either a typical 3D image (e.g. T1/T2 image) or a 4D image such as DWI, DTI, ODF and even fMRI data.

To load a Background Data, click **Load Background** in the menu or toolbar, and select a single volume image. Note that selecting multiple image is not valid here, because the program doesn’t aware which background image is laid that the bottom that might be overlaid by subsequently loaded images.

Once background image is loaded, several basic data information is shown in the data property panel.

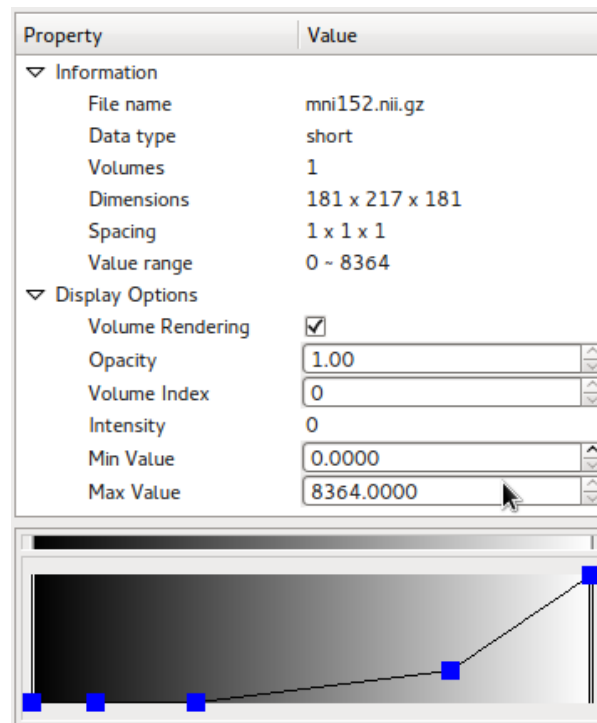


Fig. 10: property panel showing mni152 as background image

The widget below the data property panel is the color table for *Volume Rendering*. The horizontal axis indicates intensity values of a target image, while the vertical axis of the widget indicate the opacity value that maps the intensity values onto visible planes in volume rendering. One can adjust opacity map (by dragging blue opacity nodes) as well as color map to get proper rendering results.

Image Contrast Enhancement

During loading a Background image, the minimum and maximum intensity values are calculated and displayed in the data property panel. By narrow down the range of image intensity values, one can enhance image contrast.

Note that this functionality is currently applied for 2d sliced views only.

Volume Image Overlay

By loading multiple image through **Load Background** menu, one can place one image over another. Therefore, it's easy to make slice-to-slice comparison to find out image registration errors by adjusting opacity values of the upper level image.

This feature can be quite useful for verifying *Skull Stripping* results. After performing skull stripping, ...

Todo: PUT OVERLAY IMAGE HERE!

Volume Rendering

Volume rendering of the background image can be enabled in *Data Property* panel. Once volume rendering is enabled, a color table is shown to enable users to adjust color map and opacity values that maps intensity values onto visible planes. This functionality is particularly useful for visualizing skull-stripped T1 image, where the folds that increase the surface area of the cortex can be directly visible without threshold-based surface extraction.

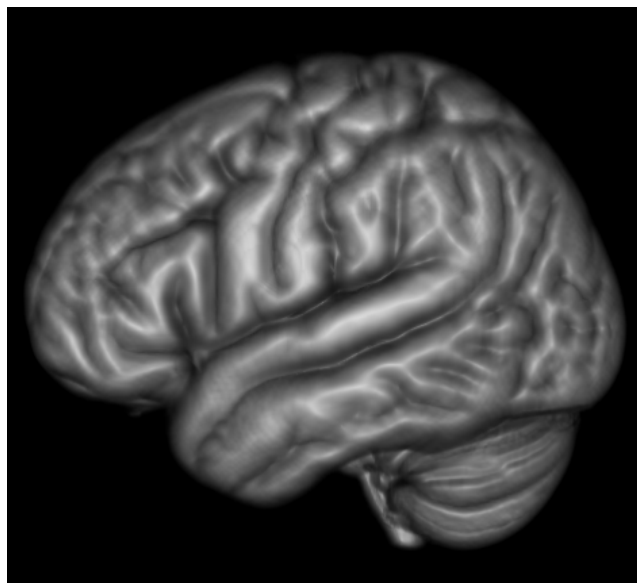


Fig. 11: volume rendering of mni152 data

Region of Interest (ROI) Layer

To describe the location of a specific region, we employ the ROI layer. It outlines surface of the region on 3d view, along with a filled area on 2d slice views.

To open a ROI file, click on ‘Load ROI’ from either menu or toolbar. Note that loading multiple ROI data files that located within the same directory is allowed. Therefore, one don’t have load splitted brain atlas files one-by-one.

To change the color of a selected ROI files, set Color to different value from the data properties panel. This is useful when more than one ROI file is loaded, and it’s hard to distinguish them by appearance.

To adjust transparency of the surface or the filled area in 2d views, set opacity value of the selected fiber file to a value range form 0 to 1.

Todo: compute the voxel-level volume size of a region, as well as its size in millimeters (by multiplying spacing).

Fiber Layer

To display deterministic tractography results, the program generates and displays .fiber/.trk files in either lines or tubes. One can load fiber files by selecting *Load Fiber* either from menu or toolbar. The definition of .fiber files is available at [Data Format](#) page. The specification of .trk file format is available at [TrackVis](#).

Property	Value
▼ Information	
Number of Fibers	180
Mean Length	112.086
Total Volume	33504
Mean FA	0.456426
Mean MD	n/a
Min Length	23.71
Max Length	99.99
▼ Display Options	
Render Type	Line
Color Code	[0,255,0]
Opacity	1.00

Fig. 12: Fiber data properties panel

Basic fiber data statistics is available on data properties panel once an arbitrary tract file is loaded. The statistics information include number of fibers in a tract, average length of fibers and total volume covered by the tract etc. These information are generated during fiber tracking process, and there is extra computation during online visualization.

Several display options are available to provided high quality rendering results. For performance reasons, tracts are rendered as slim lines by default. Therefore, detail shapes of the fibers are not clearly visible. However, location

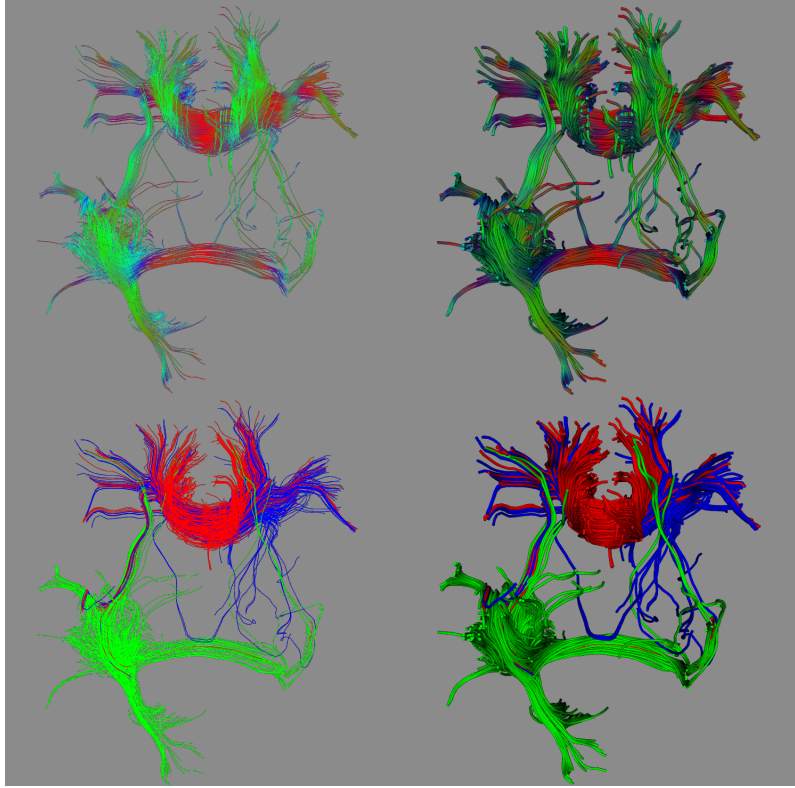


Fig. 13: A comparison of fibers rendered in different render types.

and orientation information of the lines are quite clear. By adjusting **Color Code** and **Render Type**, one can easily visualize multiple fiber files. On the above figure, two images on the left are fibers rendered as lines, while the two images on the right are rendered as tubes. Images on the top row are fibers rendered with directional coloring, while fibers on the bottom row are signed to a single color.

Note: Note that rendering whole brain fiber in ‘Tube’ rendering type can be quite slow for low performance computers. However, a progress bar would popup to show the current loading progress. Also, we recommend storing whole brain fiber files as .trk format instead of .fiber file to increase file loading efficiency.

Tensor/ODF/FOD Layer

Both diffusion tensor and ODF can be visualized within this layer. After adjusting rendering parameters, the **Render** button should be pressed in order to regenerate the scene. This prevents the time-consuming processing, every time when users are trying to change the configuration.

Diffusion tensor (DTI) file input should be a typical 4D NIFTI image, with exactly six volumes. The six values on each voxel represents entries in a 3x3 symmetric positive definite matrix. Numerical definition can be found at [DTI Reconstruction](#) section.

Diffusion orientation distribution function (ODF) file input should be a 4d image, too. However, it’s fourth dimension should be one of 15,28,45,66,91 etc. If the format is incorrect, the program would refuse to load the image to avoid potential errors. Numerical definition can be found at [SPFI Reconstruction](#) and [CSD Reconstruction](#) section.

Due to performance reasons, ODF/FOD is rendered in low resolution by default. For researchers who need high-resolution images for publication quality results, an option in the properties panel is provided.

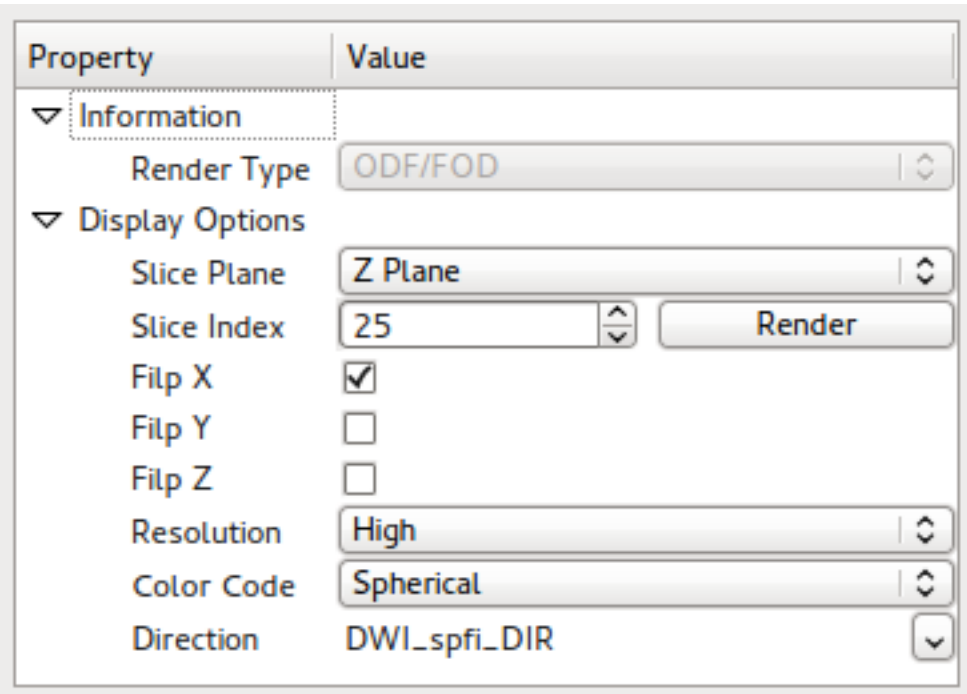


Fig. 14: Tensor/ODF/FOD data properties panel

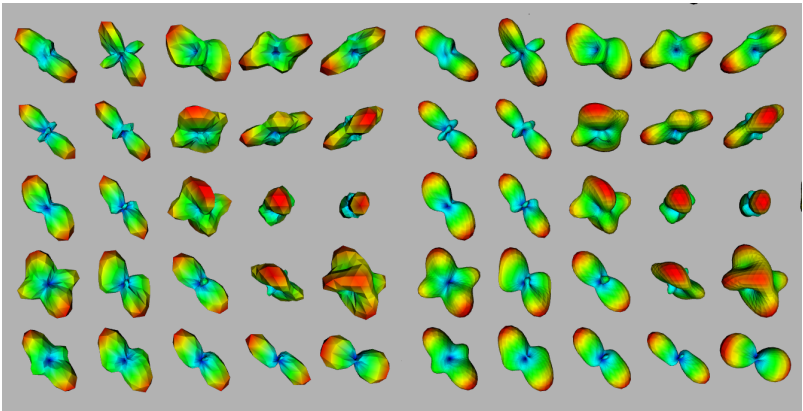


Fig. 15: A comparison of low/high resolution ODF rendering

1.1.5 Data Format

Input

Since we apply the `dcm2nii` to unify the format of input data, it can support most types of DICOM files (in folders). Please refer to the author's webpage for more information [4]. Please also get back to us if you encounter any problems.

Output

All the intermediate files are in compressed NIFTI format (.nii.gz). The final track file is in .trk format. The .trk format is from the [TrackVis](#) [17].

Briefly, header section contains following information:

Name	Data type	Bytes	Comment
id_string[6]	char	6	ID string for track file, "TRACK"
dim[3]	short	6	Dimension of the image volume.
voxel_size[3]	float	12	Voxel size of the image volume.
origin[3]	float	12	Origin of the image volume.
n_scalars	short	2	Number of scalars saved at each track point
s_name[10][20]	char	200	Name of each scalar.
n_properties	short	2	Number of properties saved at each track.
p_name[10][20]	char	200	Name of each property.
vox_to_ras[4][4]	float	64	4x4 matrix for voxel to RAS
reserved[444]	char	444	Reserved space for future version.
voxel_order[4]	char	4	Order of the original image data.
pad2[4]	char	4	Paddings.
orient_p[6]	float	24	Image orientation of the original image.
pad1[2]	char	2	Paddings.
invert_x	uchar	1	Inversion/rotation flags
invert_y	uchar	1	As above.
invert_z	uchar	1	As above.
swap_xy	uchar	1	As above.
swap_yz	uchar	1	As above.
swap_zx	uchar	1	As above.
n_count	int	4	Number of tracks stored in this track file.
version	int	4	Version number. Current version is 2.
hdr_size	int	4	Size of the header, should be 1000.

with data section in following format:

Track	Data type	Bytes	Comment
Track #1	int	4	Number of points in this track, as m
	float	(3+n_s)*4	Track Point #1.
	float	(3+n_s)*4	Track Point #2. Same as above.
	float	(3+n_s)*4	Track Point #m. Same as above.
	float	n_p*4	n_p float numbers
Track #2	Same as above.		
...			
Track #n	Same as above.		

By default, we predefined two scalars (FA,MD) and three properties (length,FA,MD) in the generated .trk file, along with

six additional values stored in `reserved` section. These are `version_num`, `num_fibers`, `mean_length`, `total_volume`, `tractFA`, `tractMD` values in `float` type (single precision).

Again, please visit [DataFormat](#) section on [TrackVis.org](#) for more detail.

1.1.6 Future Improvements

TO-DO list

1. To smooth the ODF/FOD/EAP for a smoother tract;
2. To add more efficient tracking algorithms;
3. To optimize the 3D rendering function;

1.1.7 Reference

1.2 Tutorials

1.2.1 Programming in Bash/Python

Basically the DiffusionKit is a well self-contained package which implements most of the required modules for diffusion MRI processing and analysis. Additionally, if you want to use script for batch processing for a cohort of subjects, we recommend using Python or Bash. Both of them are inherited for Linux system, and for MS/Windows one can use [Anaconda](#) and [Git-bash](#), separately.

The Python is easy to learn for basic use as a script language, although its powerful functions largely depend on 3rd party packages. To this end, we have several suggestions to begin with it. First, take a couple of hours to go through the primary Python grammars. There is a large bundle of free but kind tutorials from internet (Google “Python tutorial” or related keywords). You can choose the websites according to your preference. If you are a newbie of Python, don’t think about which version is appropriate for you and just use the latest version (Python>3.0) (In the current stage, you only need to note the difference of “print” function in different versions, which I think is not a smart change from version 3.0); and don’t waste money to buy a book since the materials from the internet are largely beyond your capacity. Several tutorial links are listed here:

For English users

1. https://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial, a short tutorial
2. <http://askpython.com/>, a short tutorial
3. <http://www.learnpython.org/>, an interactive sandbox

For Chinese users

1. <http://www.runoob.com/python3/python3-tutorial.html>, a nice tutorial
2. <http://www.cnblogs.com/vamei/archive/2012/09/13/2682778.html>, python and advanced
3. http://woodpecker.org.cn/abyteofpython_cn/chinese/, a complete reference

If you prefer to use shell script, like Bash, you can also find some entrances for tutorials. The shell script itself is easy to follow and it is a powerful tool to concatenate the underlying execution functions. It should be noted that the Bash script is only for *nix system, so it is not suitable for cross-platforms. Several tutorial links are listed here:

For English users

1. <http://linuxconfig.org/bash-scripting-tutorial>, a short introduction

2. <http://www.tldp.org/LDP/abs/html/>, a complete tutorial
3. <http://www.learnshell.org/>, an interactive sandbox

For Chinese users

1. <http://blog.jobbole.com/85183/>, a short tutorial
2. <https://serholiu.com/bash-by-example>, several simple examples
3. <http://c.biancheng.net/cpp/view/6998.html>, a complete tutorial

1.2.2 Getting Started

For your convenience, we’ve created two bash scripts the [Advanced](#) and the [Primary](#), which enable processing multiple data within a loop. The instructions for using the [Advanced](#) bash script are provided in [Download Page](#).

In this tutorial, however, we would go through the processing pipeline with the [Primary](#) bash script, which contains more features and options. It should be noted that both the [Advanced](#) and the [Primary](#) scripts can be run in simple Bash environment or emulation, since they utilize extremely basic Bash commands. So, even in MS Windows, you can run these two scripts by a Bash emulation, such as [git for Windows](#) (STRONGLY recommend), which provides a light but awesome Bash emulation. When you install this emulation, the [Primary](#) script can ben run directly once you prepared all the required data (DO remember to add the DiffusionKit installation path to your system search path). And the [Advanced](#) script need an additional Makefile-based environment.

Preprocessing

Starting from this section, we would go throught the pipeline provided in DiffusionKit, assuming DiffusionKit is properly installed following [Installation Instructions](#), along with the [Example Dataset](#) downloaded on your hard disk.

Data Format Conversion

Analyze format and Nifti format are the data formats that supported by DiffusionKit. If your image data is in DICOM format, you need to convert the data into Nifti format.

To convert DICOM format to Nifti format, please check the checkbox “DICOM to NifTI” in the main window and input the data directory that contains DICOM files.

GUI-Frontend Usage

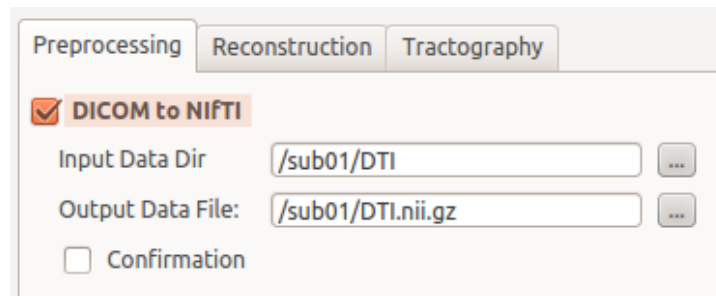


Fig. 16: Figure 1. Convert data format.

Command-Line Usage:

```
$ dcm2nii -o /path/sub01 -f DTI /path/sub01/DTI
```

Eddy Current Correction

DiffusionKit has motion correction function which is implemented with a affine registration method. To correct the effect of head motion, please check the checkbox “Eddy Current Correction” and import the original diffusion-weighted imaging data.

GUI-Frontend Usage

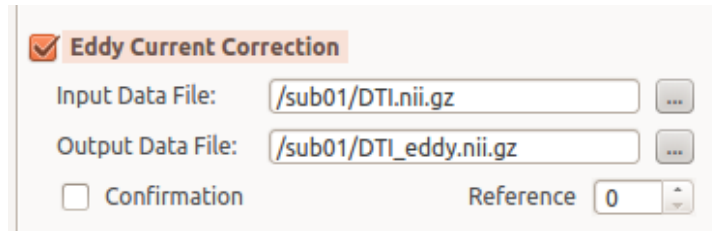


Fig. 17: Figure 2. Eddy current correction.

Command-Line Usage:

```
$ bneddy -i /path/sub01/DTI.nii.gz -o /path/sub01/DTI_eddy.nii.gz -ref 0
```

Skull Stripping

The next step is removal of extra-meningeal tissues from the MRI image of the whole head. The function of Skull Stripping is to delete non-brain tissues from an image of the whole head. An accurate brain mask will accelerate the following reconstruction.

To achieve the mask of brain, please check the checkbox “Skull Stripping” and input the image (e.g. the b0 image) you want to operate.

GUI-Frontend Usage:

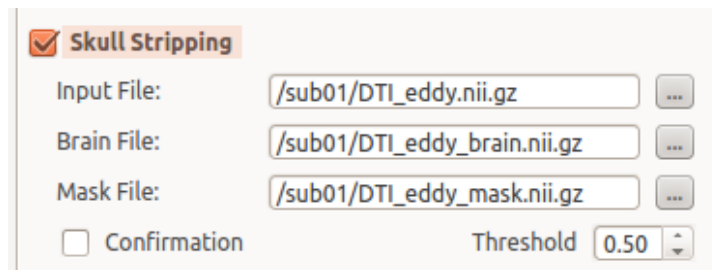


Fig. 18: Figure 3. Skull stripping.

Command-Line Usage:

```
$ bet2 /path/sub01/DTI_eddy.nii.gz /path/sub01/DTI_eddy_brain.nii.gz -m /path/sub01/DTI_eddy_mask.nii.gz -f 0.5
```

Diffusion Model Estimation

DTI Estimation

When we complete the preprocessing, we can do the job of DTI estimation. In this step, you can get diffusion tensor image and derived diffusion indexes (e.g. fractional anisotropy, mean diffusivity, radial anisotropy etc.) from the reconstruction. For GUI-frontend, please change to the “Reconstruction” label and make sure the “Output Type” is DTI. Input the files into the boxes in the “Reconstruction” Panel according to the illustrations in Figure 4.

GUI-Frontend Usage:

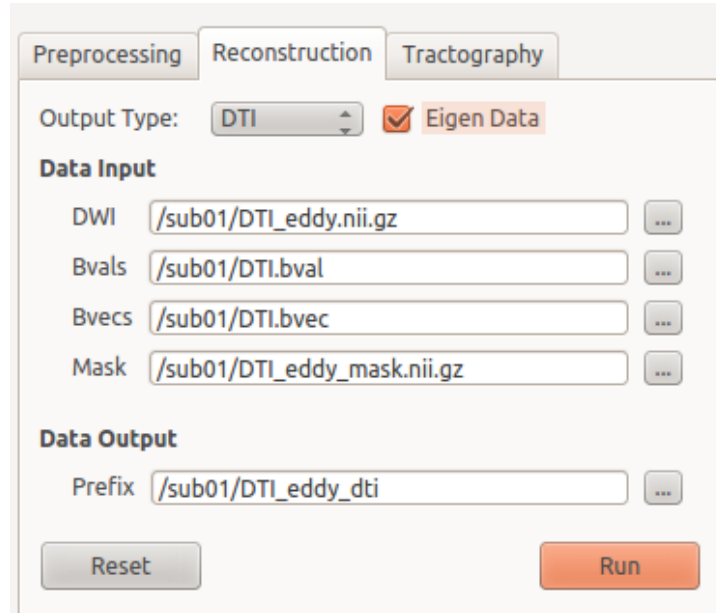


Fig. 19: Figure 4. DTI estimation.

Command-Line Usage:

```
$ bndti_estimate -d /path/sub01/DTI_eddy.nii.gz -g /path/sub01/DTI.bvec -b /path/  
→sub01/DTI.bval -m /path/sub01/DTI_eddy_mask.nii.gz -o /path/sub01/DTI_eddy_dti -  
→tensor 1 -eig 1
```

HARDI Estimation (Optional)

DiffusionKit also includes advanced HARDI reconstructions which implemented with two types of HARDI methods, the Spherical Polar Fourier Imaging (SPFI) method and Constrained Spherical Deconvolution (CSD) methods. If you have DWI data with considerable number of gradient directions (more than 45 directions) and high b-value (larger than 2000) [ref], you can apply HARDI estimation to resolve crossing fibers in the reconstruction. To call the HARDI reconstruction, change the “Output Type” to HARDI and select the HARDI method (SPFI or CSD) you want in the “Reconstruct Parameters” section. The compulsory parameters for HARDI estimation is data input and output. The example of parameters setting is illustrated in Figure 5&6.

GUI-Frontend Usage:

Command-Line Usage:

The screenshot displays the 'Reconstruction' tab of the DiffusionKit software. The 'Output Type' is set to 'HARDI'. Under 'Data Input', the fields for DWI, Bvals, Bvecs, and Mask are all populated with paths starting from '/sub01/DTI_eddy'. Under 'Data Output', the 'Prefix' field is set to '/sub01/DTI_eddy_spfi'. In the 'Reconstruct Parameters' section, the 'Method' is 'SPFI (ODF)', and the 'Generate Direction' checkbox is unchecked. The 'SH Order' is 4, 'SH Lambda' is 0.00, 'Rad Order' is 1, 'Rad Lambda' is 0.00, and 'Radius' is 0.015. The 'Generate GFA' checkbox is also unchecked. 'Reset' and 'Run' buttons are at the bottom.

Preprocessing Reconstruction Tractography

Output Type: HARDI

Data Input

DWI /sub01/DTI_eddy.nii.gz

Bvals /sub01/DTI.bval

Bvecs /sub01/DTI.bvec

Mask /sub01/DTI_eddy_mask.nii.gz

Data Output

Prefix /sub01/DTI_eddy_spfi

Reconstruct Parameters

Method SPFI (ODF) ☐ Generate Direction

SH Order 4 SH Lambda 0.00

Rad Order 1 Rad Lambda 0.00

Radius 0.015 ☐ Generate GFA

Reset Run

Fig. 20: Figure 5. HARDI estimation using SPFI method.

The screenshot displays the 'Reconstruction' tab of the DiffusionKit software. At the top, there are three tabs: 'Preprocessing', 'Reconstruction' (which is active), and 'Tractography'. Below the tabs, the 'Output Type' is set to 'HARDI'. The 'Data Input' section contains four text fields with file paths: 'DWI' (/sub01/DTI_eddy.nii.gz), 'Bvals' (/sub01/DTI.bval), 'Bvecs' (/sub01/DTI.bvec), and 'Mask' (/sub01/DTI_eddy_mask.nii.gz). Each field has a small '...' button to its right. The 'Data Output' section has a 'Prefix' field set to '/sub01/DTI_eddy_fod' with a similar '...' button. The 'Reconstruct Parameters' section includes a 'Method' dropdown set to 'CSD (FOD)', an unchecked 'Generate Direction' checkbox, and several numerical input fields: 'SH Order' (8), 'Max Iteration' (50), 'Regularization' (1.00), 'Amplitude' (0.10), 'FA Range' (0.75), and 'Resolution' (300). At the bottom, there are 'Reset' and 'Run' buttons.

Preprocessing Reconstruction Tractography

Output Type: HARDI

Data Input

DWI /sub01/DTI_eddy.nii.gz ...

Bvals /sub01/DTI.bval ...

Bvecs /sub01/DTI.bvec ...

Mask /sub01/DTI_eddy_mask.nii.gz ...

Data Output

Prefix /sub01/DTI_eddy_fod ...

Reconstruct Parameters

Method CSD (FOD) ☐ Generate Direction

SH Order 8 FA Range 0.75

Max Iteration 50 0.95

Regularization 1.00 Resolution 300

Amplitude 0.10

Reset Run

Fig. 21: Figure 6. HARDI estimation using CSD method.

```
$ bnhardi_ODF_estimate -d /path/sub01/DTI_eddy.nii.gz -g /path/sub01/DTI.bvec -b /
↳path/sub01/DTI.bval -m /path/sub01/DTI_eddy_mask.nii.gz -o /path/sub01/DTI_eddy_spfi
$ bnhardi_FOD_estimate -d /path/sub01/DTI_eddy.nii.gz -g /path/sub01/DTI.bvec -b /
↳path/sub01/DTI.bval -m /path/sub01/DTI_eddy_mask.nii.gz -o /path/sub01/DTI_eddy_fod
```

Tractography

DiffusionKit provides the function of fibertracking which is implemented with a deterministic streamline method. You can perform tractography using the diffusion tensors reconstructed with DTI or diffusion/fiber ODFs reconstructed with SPFI/CSD. To call the function of fibertracking, change the “Processing” Panel to “Tractography”. Select the data type (DTI or HARDI) according to the data you used (tensor or diffusion/fiber ODF). Input the FA map acquired in DTI estimation to the box of “FA”. Provide the seeds image based on your research. For instance, if you want to do whole brain fibertracking, you can input a whole brain mask in the box of “Seeds”. You can adjust other tracking parameters to satisfy your study. Please refer to the illustration in Figure 7&8 for more details.

GUI-Frontend Usage

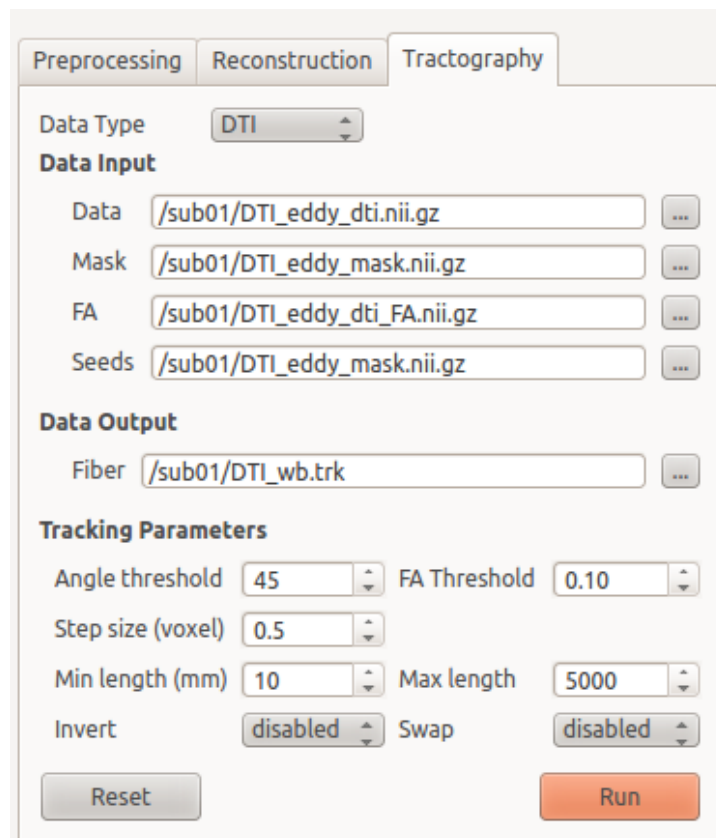


Fig. 22: Figure 7. Fibe tracking based on DTI.

Command-Line Usage

```
$ bndti_tracking -d /path/sub01/DTI_eddy_dti.nii.gz -m /path/sub01/DTI_eddy_mask.nii.
↳gz -s /path/sub01/DTI_eddy_mask.nii.gz -fa /path/sub01/DTI_eddy_dti_FA.nii.gz -o /
↳path/sub01/DTI_wb.trk
$ bnhardi_tracking -d /path/sub01/DTI_eddy_dti.nii.gz -m /path/sub01/DTI_eddy_mask.
↳nii.gz -s /path/sub01/DTI_eddy_mask.nii.gz -fa /path/sub01/DTI_eddy_dti_FA.nii.gz -
↳o /path/sub01/DTI_wb.trk
```

(continues on next page)

The image shows a software interface for fiber tracking, specifically the 'Tractography' tab. It features several input fields for data files, output file, and various tracking parameters. The 'Data Type' is set to 'HARDI'. The 'Data Input' section includes fields for 'Data', 'Mask', 'FA', and 'Seeds', all pointing to files in the '/sub01/DTI_eddy' directory. The 'Data Output' section has a 'Fiber' field pointing to '/sub01/HARDI_wb.trk'. The 'Tracking Parameters' section includes sliders for 'Angle threshold' (45), 'FA Threshold' (0.10), 'Step size (voxel)' (0.5), 'Min length (mm)' (10), and 'Max length' (5000). There are also dropdown menus for 'Invert' (disabled) and 'Swap' (disabled). At the bottom, there are 'Reset' and 'Run' buttons.

Preprocessing Reconstruction **Tractography**

Data Type **HARDI**

Data Input

Data ...

Mask ...

FA ...

Seeds ...

Data Output

Fiber ...

Tracking Parameters

Angle threshold FA Threshold

Step size (voxel)

Min length (mm) Max length

Invert Swap

Fig. 23: Figure 8. Fiber tracking based on HARDI.

(continued from previous page)

Construction of brain networks

When we get connectivity data from tractography, we can consider construction of brain networks. A network is a collection of nodes and links (edges) between pairs of nodes. In structural brain networks, we select specific ROIs of brain as nodes and the number of fibers which end in the pair of ROIs or other derived diffusion indices as edges. DiffusionKit provides the function of `bn_network` to construct brain network using given ROIs and whole brain tractography achieved from last step. Only command-line usage is available for `bn_network`. A text file which contains the paths and ROI filenames is needed to input when performing `bn_network`. Please refer to the example script and example data for details.

Command-Line Usage

```
$ bn_network -fiber /path/sub01/DTI_wb.trk -roi /path/sub01/roi.txt -outfiber 1 -o /
↳ path/sub01/network.txt
```

1.2.3 Generate a ROI file from a given ROI list

If you want to generate a `roi.nii.gz` file in the space of a reference `ref.nii.gz`, with a given list of ROI coordinates and according weight value, where the radii are unified by the input radius `r`, you can call the `bncalc` function in-loop as the file `gen_ROI_from_list.sh`. The content is as following:

```
#!/bin/bash
# This is to call the bncalc function to create a .nii.gz mask image
# from a list of ROI centers and radii
# Jan 7, 2016, by NMZUO

# The list.txt has the format as:
# x      y      z      weightValue # this could be used for color; set 1 by default
# -2     38     36     3.6
# 65     -31     -9     -2.7
# 12     36     20     4
# .....

Usage() {
    echo ""
    echo "This is to call the bncalc function to create"
    echo "a .nii.gz mask image from a list of ROI centers."
    echo "Usage:"
    echo "bash gen_ROI_from_list.sh list.txt r ref.nii.gz outmask.nii.gz"
    echo -e "    list.txt\t\t\t The input list containing ROI coordinates (MNI mm)"
    echo -e "    r\t\t\t\t\t The radius (mm) of each ROI"
    echo -e "    ref.nii.gz\t\t The reference image where the ROIs stay"
    echo -e "    outmask.nii.gz\t The output .nii.gz image"
}

if [ $# -lt 3 ]; then
    Usage
    exit 1
fi

tmpfile=${$_}tmp_
```

(continues on next page)

(continued from previous page)

```

iCount=0
while read line; do
    roi=($line)
    idx=`printf "%06d" $iCount`
    bncalc -i $3 -roi ${roi[0]}, ${roi[1]}, ${roi[2]}, $2 -o ${tmpfile}${idx}.nii.gz
    if [ ${#roi[@]} -gt 3 ]; then
        bncalc -i ${tmpfile}${idx}.nii.gz -mul ${roi[3]} -o ${tmpfile}${idx}.nii.gz
    fi
    iCount=$((iCount+1))
done < $1

cp ${tmpfile}"000000".nii.gz $4
for i in `seq 1 $((iCount-1))`
do
    echo -e "$i \c"
    idx=`printf "%06d" $i`
    if [ ${#roi[@]} -lt 4 ]; then
        bncalc -i ${tmpfile}${idx}.nii.gz -mul $((i+1)) -o ${tmpfile}${idx}.nii.gz
    fi
    bncalc -i $4 -add ${tmpfile}${idx}.nii.gz -o $4
done
echo "Done!"
rm -f ${tmpfile}?????.nii.gz

```

1.2.4 Map ROIs defined in one space to another individual space

This is an example to map the ROI definitions (in the initial space, e.g. MNI 2mm space) to an individual space. For example, the source folder could be:

```

nmzuo@:dp_classify$ tree
.
├── bet_reg.sh
├── bet_reg_sub2std.sh
├── copy_data_from_231.sh
├── DP
│   ├── DP_0001
│   └── DP_0002
└── .....

```

where the fold for each subject could be:

```

nmzuo@:DP_0001$ tree
.
├── dwi
│   ├── bval
│   ├── bvec
│   ├── bvec_rotated
│   ├── DP_0001_dwi_eddy_b0.nii.gz
│   ├── DP_0001_dwi_eddy.nii.gz
│   ├── DP_0001_dwi_mask.nii.gz
│   ├── DP_0001_dwi.nii.gz
│   ├── dti_FA.nii.gz
│   ├── dti_L1.nii.gz
│   └── dti_L2.nii.gz

```

(continues on next page)

(continued from previous page)

```

├── dti_L3.nii.gz
├── dti_MD.nii.gz
├── dti.nii.gz
├── dti_RA.nii.gz
├── dti_V1.nii.gz
├── dti_V2.nii.gz
├── dti_V3.nii.gz
└── t1
    ├── DP_0001_t1_brain_mask.nii.gz
    ├── DP_0001_t1_brain.nii.gz
    └── DP_0001_t1.nii.gz

```

and *list_dp.txt* is the name list of the subjects as (the column other than the first one is the possible comment for the subject):

```

nmzuo@:dp_classify$ cat list_dp.txt
DP_0001
DP_0002 #no cerebellum
.....

```

Then the following *bash code* is able to map the ROI in the standard space to the subject space (for each subject in *list_dp.txt*). Certainly the initial ROI does not require to be in the standard space in this code.

```

atlas='/datc/software/BrainnetomeAtlas/BNAtlas_246_2mm.nii.gz' # Brainnetome Atlas
↳ in MNI space
mni='/datc/software/fsl5.0/data/standard/MNI152_T1_2mm_brain.nii.gz'

for dat in `cat list_dp.txt |awk '{print $1}'` ` #only read the first column
do
    echo $dat
    cpath='/datd/dp_classify'
    cpath=$cpath/${dat%%_*}/${dat}/
    oldp=`pwd`
    cd $cpath
    echo 'aladin'
    reg_aladin -ref t1/$dat'_t1_brain.nii.gz' -flo $mni -aff aff.txt -voff
    echo 'f3d'
    reg_f3d -ref t1/$dat'_t1_brain.nii.gz' -flo $mni -aff aff.txt -cpp cpp.nii.gz
↳ -maxit 3 -ln 2 -voff
    rm -f aff.txt
    echo 'aladin'
    reg_aladin -ref dwi/$dat'_dwi_eddy_b0.nii.gz' -flo t1/$dat'_t1_brain.nii.gz' -
↳ aff aff0.txt -voff

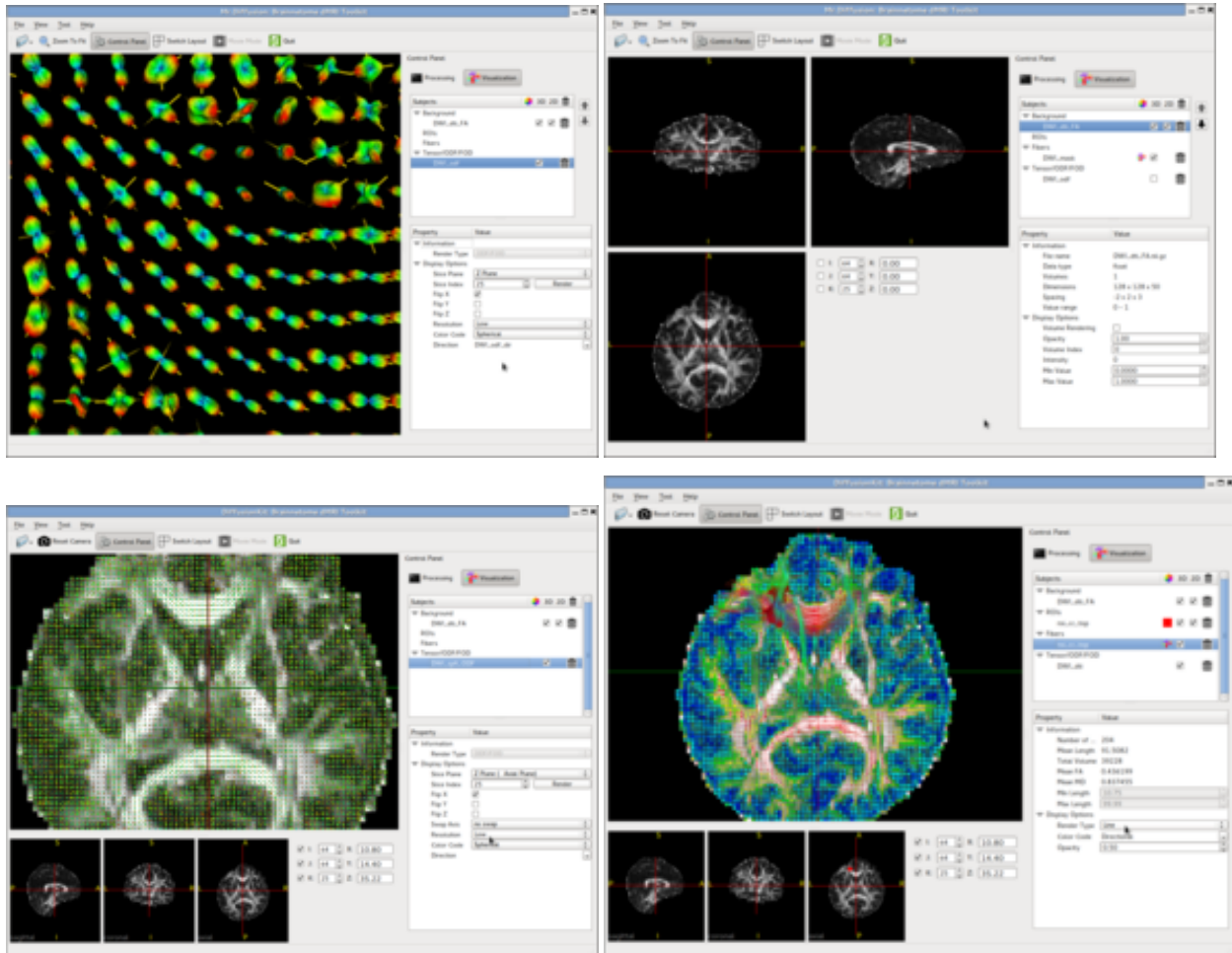
    reg_resample -ref t1/$dat'_t1_brain.nii.gz' -flo $atlas -trans cpp.nii.gz -
↳ inter 0 -res jhu_t1.nii.gz -voff
    reg_resample -ref dwi/$dat'_dwi_eddy_b0.nii.gz' -flo jhu_t1.nii.gz -trans aff0.
↳ txt -inter 0 -res jhu_b0.nii.gz -voff
    rm -f aff0.txt cpp.nii.gz outputResult.nii jhu_t1.nii.gz
    mv jhu_b0.nii.gz dwi/atlas_b0.nii.gz
    cd $oldp
done

```

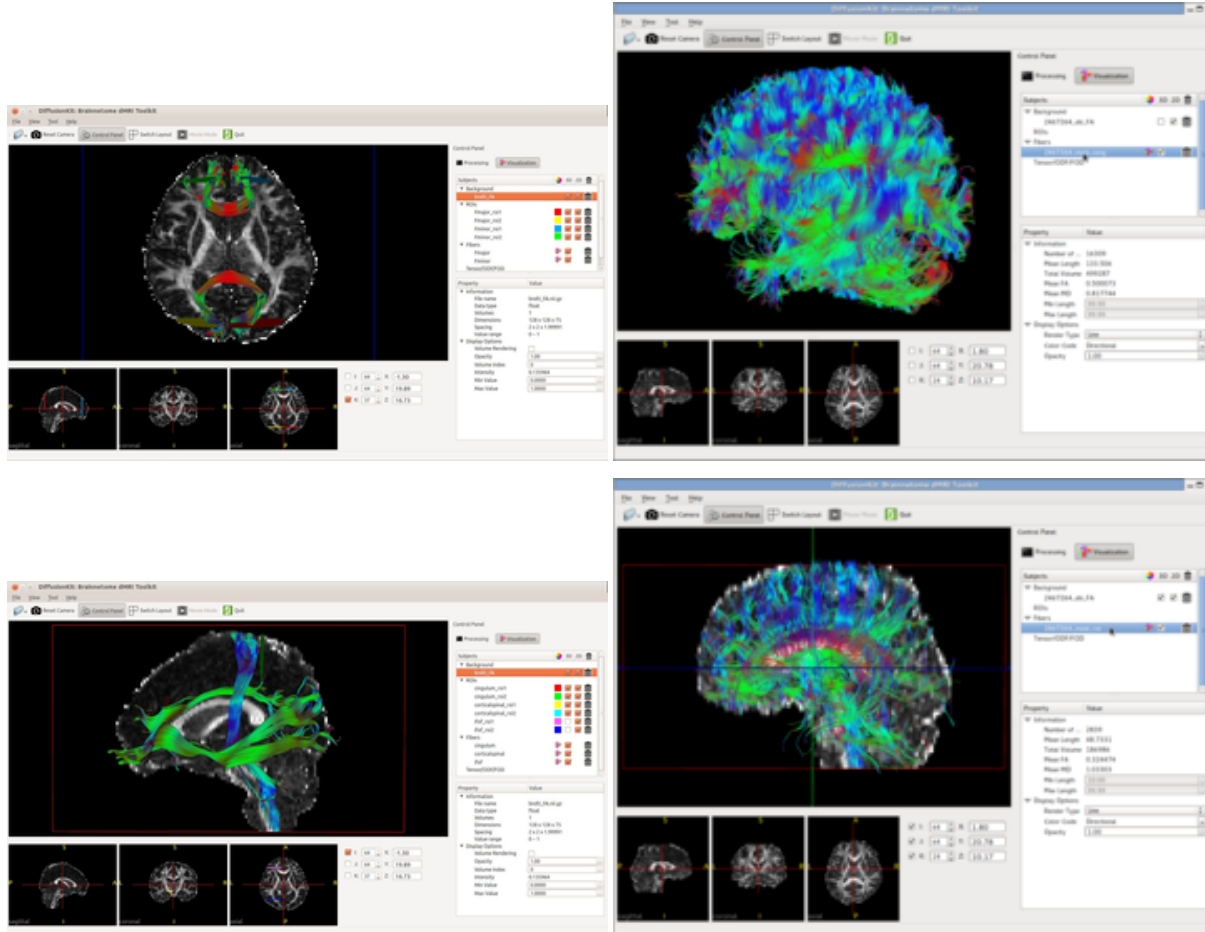
1.3 Screenshots

1.3.1 Tensor and ODF/FOD Visualization

[click to see high resolution images]



1.3.2 Tractography Visualization



1.4 Download and Install

1.4.1 Download Links

Latest Release (Sept. 28, 2018)

Previous Release

See [Releases](#) for our previously compiled packages.



Windows Installer (x86-64) [14 MB]



Windows (x86-64) (Extract then use. No installation required) [41 MB]



Linux Binary Package (x86-64) [50 MB]



Windows Installer (x86-64) [14 MB]



Windows (x86-64) (Extract then use. No installation required) [41 MB]



Linux Binary Package (x86-64) [50 MB]

Example Data and Test Script

- List File: `list.txt` [12 Bytes]
- Subject 01: `sub01.tar.gz` [87.1 MB]
- Subject 02: `sub02.tar.gz` [85.5 MB]
- Brain Atlas: `atlas.tar.gz` [1.57 MB]
- Test Script [`process_advanced.sh`] [`process_primary.sh`]
- List File: `list.txt` [12 Bytes]
- Subject 01: `sub01.tar.gz` [87.1 MB]
- Subject 02: `sub02.tar.gz` [85.5 MB]
- Brain Atlas: `atlas.tar.gz` [1.57 MB]
- Test Script [`process_advanced.sh`] [`process_primary.sh`]

And this is what we have in the data above:

```
— atlas
  — aal.nii.gz
  — aal.nii.lut
  — aal.nii.txt
  — aal_roi_024.txt
  — aal_roi_090.txt
  — BN_Atlas_246_1mm.nii.gz
  — BN_Atlas_246.txt
  — ch2bet.nii.gz
— atlas.tar.gz
```

(continues on next page)

(continued from previous page)

```

├── list.txt
├── process_advanced.sh
├── process_primary.sh
├── sub01
│   ├── dwi.bval
│   ├── dwi.bvec
│   ├── dwi.nii.gz
│   └── t1.nii.gz
├── sub01.tar.gz
├── sub02
│   ├── dwi.bval
│   ├── dwi.bvec
│   ├── dwi.nii.gz
│   └── t1.nii.gz
└── sub02.tar.gz

```

We recommend users to follow the [Tutorial Page](#) for a step-by-step introduction of the functions within DiffusionKit.

IN A HURRY? Download all the data files (REQUIRED) above and run the Bash script `process_advanced.sh` or `process_primary.sh`. These two scripts have exactly the same functions, except that the primary one is for those who don't have too much background in Bash scripting and the advanced one is Makefile-based which avoids repeatedly compiling the unchanged data in multiple compiles. Or simply run the following commands to do everything (for Linux ONLY).

```

# install DiffusionKit
kitv='v1.4-r161127' ## This line should be changed according to the version you
↳desired.
kitname='DiffusionKitSetup-x86_64-${kitv}'
wget https://github.com/brainnetome/diffusionkit/releases/download/${kitv}/${kitname}
↳'.tar.gz'
tar zxvf ${kitname}'.tar.gz'
export PATH=$PATH:$(pwd)/${kitname}/bin

# get the data and run!
wget https://github.com/brainnetome/diffusionkit/releases/download/v1.4-r161127/list.
↳txt
wget https://github.com/brainnetome/diffusionkit/releases/download/v1.4-r161127/sub01.
↳tar.gz
wget https://github.com/brainnetome/diffusionkit/releases/download/v1.4-r161127/sub02.
↳tar.gz
wget https://github.com/brainnetome/diffusionkit/releases/download/v1.4-r161127/atlas.
↳tar.gz
wget https://raw.githubusercontent.com/brainnetome/diffusionkit/master/source/static/
↳process_advanced.sh
chmod +x process_advanced.sh
./process_advanced.sh

```

```

# install DiffusionKit
kitv='v1.3-r160923' ## CAUTION: This line should be changed according to the version.
kitname='DiffusionKitSetup-x86_64-${kitv}'
wget http://ddl.escience.cn/ff/emKy?func=download -O ${kitname}'.tar.gz'
tar zxvf ${kitname}'.tar.gz'
export PATH=$PATH:$(pwd)/${kitname}/bin

# get the data and run!
wget https://github.com/brainnetome/diffusionkit/releases/download/v1.3-r160923/list.
↳txt

```

(continues on next page)

(continued from previous page)

```
wget http://ddl.escience.cn/ff/emBl?func=download -O sub01.tar.gz
wget http://ddl.escience.cn/ff/emBm?func=download -O sub02.tar.gz
wget http://ddl.escience.cn/ff/emBo?func=download -O atlas.tar.gz
wget https://raw.githubusercontent.com/brainnetome/diffusionkit/master/source/static/
→process_advanced.sh
chmod +x process_advanced.sh
./process_advanced.sh
```

And more simpler way is to copy the code in the above panel (choose the server close to you) and save it as a file, such as [test-diffusionkit.sh](#), or [test-diffusionkit-china.sh](#). and run to take a quick look how DiffusionKit can **BUILD A BRAIN NETWORK JUST A FEW MINUTES!**

1.4.2 System requirement

Basically this software can run in any system, including 64-bit MS Windows/Linux OS, although currently we only tested and released the binary packages for Windows/Linux OS. The software is developed based on C/C++, and some platform independent packages, including VTK, and OpenCV, etc. However, for high-performance data processing and visualization, we recommend using 64-bit OS with multi-core CPU and standalone graphics card.

1.4.3 Install/Uninstall

Please download the package from <http://diffusion.brainnetome.org> , according to your own OS. Unpack the files to where you want and you can enjoy the software. The 64-bit OS is recommended for high-performance data processing. Each installation package is completely standalone so you DO NOT need to install ANY other dependency to run the software. If you encounter any dependency problem please DO [contact us](#).

For MS Windows OS

Double click the `DiffusionKitSetup-WIN64-v1.4-r161127.exe` file and then choose the destination path according to the wizard. You may need to provide administrator permission if you want to put the files into the system path. Similarly, to uninstall you only need to hit the menu of “uninstall” in the MS Windows start menu.

For Linux OS

Glibc>=2.2 is required. Download the `DiffusionKitSetup-x86_64-v1.4-r161127.tar.gz`, and then

```
tar zxvf DiffusionKitSetup-x86_64-v1.4-r161127.tar.gz
export PATH=$PATH:`pwd`/DiffusionKitSetup-x86_64-v1.4-r161127/bin
```

You could add the path to the \$PATH in the `~/.bashrc` file, by adding the following line,

```
export $PATH=$PATH:/your/path/to/diffusionkit
```

To uninstall the software, just manually remove the entire folder where you untar-ed the .tar.gz file.

The citation for DiffusionKit could be:

Sangma Xie, Liangfu Chen, Nianming Zuo and Tianzi Jiang, **DiffusionKit: A Light One-Stop Solution for Diffusion MRI Data Analysis**, *Journal of Neuroscience Methods*, vol. 273, pp. 107-119, 2016. [\[PDF\]](#) .

1.5 License

Copyright © 2012-2016 [Brainnetome Center](#), CASIA

This software is provided ‘as-is’, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to use this software for any purpose, including research purpose and commercial applications, subject to the following restrictions:

1. The origin of this software must not be misrepresented; If you use this software in your work, an acknowledgment is required. A written permission is also required for commercial use.
2. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. All the adopted third party modules keep their own copyright.
4. This notice may not be removed or altered from any distribution.

1.5.1 The licenses of the third party toolboxes

BET (Brain Extraction Tool) v2.1 - FMRIB Analysis Group, Oxford

FMRIB Software Library, Release 5.0 (c) 2012, The University of Oxford (the ‘Software’)

The Software remains the property of the University of Oxford (‘the University’).

The Software is distributed ‘AS IS’ under this Licence solely for non-commercial use in the hope that it will be useful, but in order that the University as a charitable foundation protects its assets for the benefit of its educational and research purposes, the University makes clear that no condition is made or to be implied, nor is any warranty given or to be implied, as to the accuracy of the Software, or that it will be suitable for any particular purpose or for use under any specific conditions. Furthermore, the University disclaims all responsibility for the use which is made of the Software. It further disclaims any liability for the outcomes arising from using the Software.

The Licensee agrees to indemnify the University and hold the University harmless from and against any and all claims, damages and liabilities asserted by third parties (including claims for negligence) which arise directly or indirectly from the use of the Software or the sale of any products based on the Software.

No part of the Software may be reproduced, modified, transmitted or transferred in any form or by any means, electronic or mechanical, without the express permission of the University. The permission of the University is not required if the said reproduction, modification, transmission or transference is done without financial return, the conditions of this Licence are imposed upon the receiver of the product, and all original and amended source code is included in any transmitted product. You may be held legally responsible for any copyright infringement that is caused or encouraged by your failure to abide by these terms and conditions.

You are not permitted under this Licence to use this Software commercially. Use for which any financial return is received shall be defined as commercial use, and includes (1) integration of all or part of the source code or the Software into a product for sale or license by or on behalf of Licensee to third parties or (2) use of the Software or any derivative of it for research with the final aim of developing software products for sale or license to a third party or (3) use of the Software or any derivative of it for research with the final aim of developing non-software products for sale or license to a third party, or (4) use of the Software to provide any service to an external organisation for which payment is received. If you are interested in using the Software commercially, please contact Isis Innovation Limited (‘Isis’), the technology transfer company of the University, to negotiate a licence. Contact details are: innovation@isis.ox.ac.uk quoting reference BS/9564.

dcm2nii - Chris Rorden's dcm2niix, version 24Nov2014

The Software has been developed for research purposes only and is not a clinical tool Copyright (c) 2014-2016 Chris Rorden. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright owner nor the name of this project (dcm2niix) may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT OWNER 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NiftyReg v1.3.9 - University College London, UK

Copyright (c) 2009, University College London, United-Kingdom All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the University College London nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.6 FAQs

1.6.1 1. Where can I download example data?

We provide a group of [test data](#) to test the software, and the data was also used in [\[11\]](#).

1.6.2 2. Why images and fibers are not properly aligned in my case?

This typically happens when nifti image headers are modified before generating fiber.

In DiffusionKit, a command named `bninfo` is provided to print out the header information inside a NIFTI image. To check whether an NIFTI data is canonical:

```
$ bninfo data.nii.gz
```

you would get:

```
...
qform_i_orientation = 'Left-to-Right'
qform_j_orientation = 'Posterior-to-Anterior'
qform_k_orientation = 'Inferior-to-Superior'
...
```

This indicate that you are lucky. And the matrix that transforms voxels coordinates (I/J/K) to physical locations (X/Y/Z) is orthogonal. However, it doesn't matter if this isn't true, the image can be converted with `bnconvert`. With `--reorient` argument, it transforms the input image into orthogonal one with head information modified as well.

```
$ bnconvert data_brain.nii.gz data_brain_r.nii.gz -r
```

Warning: However, there would be problems if `bnconvert` converts a 4D vector image, e.g. tensor data, ODF/FOD data etc.

Usually, the reorient step should be taken when DICOM image is converted to NIFTI image, with argument `-r y`.

For more information, we recommend reading FSL's [Orientation Explained](#) and [Docs on qform and sform](#) to get a good understanding of orientation layout within Nifti data format.

1.6.3 3. How to extract the gradient table if the dcm2nii/MRlcron fails?

The `dcm2nii`/`MRlcron` are useful to extract the gradient table for the DWI data. Occasionally, they fails to get the gradient table, though it indeed get the NIFTI data from the DICOM images. Herein we provide a temporary solution by calling the `MATLAB/dicominfo` which depends on a specified keyword dictionary.

```
function findGrad(dcmfile)
% This script is special to locate the individual grad directions
% for each DICOM file, if the dicom2nii tool failed to extract them.
% by NMZUO, Sept. 12, 2014

% Change the following if still fails
%prestr = 'DiffusionGradientOrientation';
prestr = 'DiffusionGradient';

myhdr = dicominfo(dcmfile);
myfind = fieldnamesr(myhdr, prestr);
iLen = length(myfind);
iCount = 0;

for i=1:iLen
    mystr = myfind{i};
```

(continues on next page)

(continued from previous page)

```
if strfind(mystr, prestr)
    iCount = iCount + 1;
    disp(['Field containing ' prestr ': ' num2str(iCount) ]);
    disp(mystr);

    % to extract the grad directions
    eval(strcat('myhdr.', mystr))
end
end
end
```

Please download the two files [fieldnamesr.m](#) and [findGrad.m](#) . The first one is to generate the full string of the keyword dictionary and second is to find the locations. If your problem remains unsolved, please [send us](#) your data (only one 2D/3D DICOM image is enough) and then we will update our DICOM dictionary.

1.7 Support

1.7.1 Feedback

If you encounter any problems or have any suggestions, please feel free to get back to us, diffusion.kit@nlpr.ia.ac.cn .

Note: Document last updated on Apr 27, 2020.
