

Type of article: Research Paper

DiffusionKit: A Light One-Stop Solution for Diffusion MRI Data Analysis

Sangma Xie^{1,2}, Liangfu Chen^{1,2}, Nianming Zuo^{1,2} and Tianzi Jiang^{1,2,3,4,5}

- (1) Brainnetome Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (2) National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (3) CAS Center for Excellence in Brain Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (4) Queensland Brain Institute, University of Queensland, Brisbane, QLD 4072, Australia
- (5) Key Laboratory for NeuroInformation of the Ministry of Education, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author:

Nianming Zuo can be contacted at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; Tel.: +86 10 8254 4768; Fax: +86 10 8254 4447; E-mail: nmzuo@nlpr.ia.ac.cn

HIGHLIGHTS

- DiffusionKit has a full pipeline for (pre-)processing and visualization of diffusion MRI data.
- DiffusionKit has cross-platform support and a small installation size without 3rd party dependency.
- DiffusionKit has both a GUI interface and command-line functions that enable easy operation and batch processing.

DiffusionKit: A Light One-Stop Solution for Diffusion MRI Data Analysis

Sangma Xie^{1,2}, Liangfu Chen^{1,2}, Nianming Zuo^{1,2} and Tianzi Jiang^{1,2,3,4,5}

- (1) Brainnetome Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (2) National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (3) CAS Center for Excellence in Brain Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
- (4) Queensland Brain Institute, University of Queensland, Brisbane, QLD 4072, Australia
- (5) Key Laboratory for NeuroInformation of the Ministry of Education, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author:

Nianming Zuo can be contacted at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; Tel.: +86 10 8254 4768; Fax: +86 10 8254 4447; E-mail: nmzuo@nlpr.ia.ac.cn

HIGHLIGHTS

- DiffusionKit has a full pipeline for (pre-)processing and visualization of diffusion MRI data.
- DiffusionKit has cross-platform support and a small installation size without 3rd party dependency.
- DiffusionKit has both a GUI interface and command-line functions that enable easy operation and batch processing.

Abstract (250 words limit)

Background: Diffusion magnetic resonance imaging (dMRI) techniques are receiving increasing attention due to their ability to characterize the arrangement map of white matter *in vivo*. However, the existing toolkits for dMRI analysis that have accompanied this surge possess noticeable limitations, such as large installation size, an incomplete pipeline, and a lack of cross-platform support.

New Method: In this work, we developed a light, one-stop, cross-platform solution for dMRI data analysis, called **DiffusionKit**. It delivers a complete pipeline, including data format conversion, dMRI preprocessing, local reconstruction, white matter fiber tracking, fiber statistical analyses and various visualization schemes. Furthermore, DiffusionKit is a self-contained executable toolkit, without the need to install any other software.

Results: The DiffusionKit package is implemented in C/C++ and Qt/VTK, is freely available at <http://diffusion.brainnetome.org> and <https://www.nitrc.org/projects/diffusionkit>. The website of DiffusionKit includes test data, a complete tutorial and a series of tutorial examples. A mailing list has also been established for update notification and questions and answers.

Comparison with Existing Methods: DiffusionKit provides a full-function pipeline for dMRI data analysis, including data processing, modeling and visualization. Additionally, it provides both a graphical user interface (GUI) and command-line functions, which are helpful for batch processing. The standalone installation package has a small size and cross-platform support.

Conclusions: DiffusionKit provides a complete pipeline with cutting-edge methods for dMRI data analysis, including both a GUI interface and command-line functions. The rich functions for both data analysis and visualization will facilitate and benefit dMRI research.

Keywords: Diffusion MRI; DTI; HARDI; Anatomical connectivity; DiffusionKit

1. Introduction

Brain network studies have become a significant approach for understanding how different functional parcellations of the brain interact with each other. Currently, there are two critical methods to inspect brain networks, functional networks and anatomical networks ([Jiang, 2013](#)), in which the anatomical networks consist of the neural bases and physical connections for the functional networks ([Sporns et al., 2005](#)). To date, diffusion magnetic resonance imaging (diffusion MRI, also known as dMRI) has been recognized as an irreplaceable technique to investigate the anatomical connectivity of the brain *in vivo*, and based on the data collected by dMRI, mathematical models have been applied to characterize the distribution of the water molecules constrained by the white matter microstructure ([Zuo et al., 2012](#)).

The brain connectivity and network analyses based on dMRI data consist of a series of image processing and modeling steps ([Sotiropoulos et al., 2013](#)). First, each separate volume of the diffusion-weighted imaging (DWI) series is corrected for the distortions induced by eddy-current and susceptibility, both of which are potential sources of off-resonance field, and for head motion, which blurs the images and affects the image alignment ([Le Bihan et al., 2006](#)). Additionally, to avoid processing of the area outside of the brain tissue, a skull stripping step is strongly recommended. Second, for each of the voxels, the diffusion model is estimated from the DWI series. According to the sampling scheme in the q-space, there are a variety of estimation strategies to dissect the white matter propagation within a voxel. The most traditional and popular strategy is diffusion tensor imaging (DTI) ([Basser et al., 1994](#); [Le Bihan et al., 2001](#)). Recently, for the deciphering of crossing fibers, the decomposition-based spherical polar Fourier imaging (SPFI) method and the deconvolution-based constrained spherical deconvolution (CSD) method have emerged as two representative families of high angular resolution diffusion imaging (HARDI) techniques ([Xie et al., 2015](#)). Thus, voxel-based analysis and tract-based spatial statistics (TBSS) ([Smith et al., 2006](#)) can be performed, such as for fractional anisotropy (FA), mean diffusivity (MD), relative anisotropy (RA) ([Assaf and Pasternak, 2008](#); [Le Bihan et al., 2001](#)). Third, once maps of white matter distribution within voxels are obtained, the white matter fiber connections can be tracked by a number of tractography approaches ([Mori and van Zijl, 2002](#)). Finally, to construct the connectivity between two brain areas, specific attributes can be derived from the connecting fiber bundles, such as the mean fractional anisotropy (FA), mean diffusivity (MD), number of fibers, and volumes occupied by the fiber bundles ([Sporns, 2011](#); [Sporns et al., 2005](#)). The researchers who want to implement the entire pipeline should be acquainted with corresponding algorithms and proficient in programming.

Meeting the requirements of researchers from multidisciplinary backgrounds, a large number of toolkits have been developed, including data preprocessing ([FSL](#); [SPM](#)) and diffusion tensor or orientation distribution function modeling ([FSL](#); [MRtrix](#); [TrackVis](#)), fiber tracking and visualization ([DTITool](#); [TrackVis](#)). However, these tools have mainly focused on specific steps, and they do not provide a full set of components, consisting of data conversion, data (pre-)processing, fibertracking, network construction, image/fiber view and visualization. Recently, to address these drawbacks, a few integrated tools have been developed, such as PANDA ([Cui et al., 2013](#)), ExploreDTI ([Leemans et al., 2009](#)), MedInria ([Toussaint et al.,](#)

2007), Dipy ([Garyfallidis et al., 2014](#)) and MIPAV ([MIPAV](#)). For a more detailed survey, please see the Table. 1. However, these new tools have their own limitations. First, some of them are MATLAB (The Mathworks, Inc.) based. MATLAB, an interpreted language, will execute more slowly than compiled languages ([Chapman, 2015](#)) and is not powerful enough in rendering surface of the brain and numerous fibers. Second, a large number of third-party dependencies for installation can result in a large software size. Third, there is no complete pipeline or cross-platform support.

In this work, we developed a light but comprehensive solution, called DiffusionKit, for dMRI data (pre-)processing and visualization. The remainder of the manuscript is organized as follows. In the Material and methods section, we introduce the overall design of DiffusionKit and the functions of the main modules with explanations of the principles and implementations. Then, results and some examples are presented in the Results and discussion section to validate the given functions. In the Conclusions section, the main modules and features are summarized, and comparisons with existing tools are also tabulated.

2. Material and methods

2.1 Overall design

The overall design framework is illustrated in Fig. 1. DiffusionKit was developed as a cross-platform framework, using C/C++ for computation, VTK ([VTK](#)) for visualization, and Qt for graphical user interface (GUI) design. Both GPU and CPU computing have been implemented to achieve a high frame rate for rendering complex scenes, particularly whole-brain tractography. The project was managed using the compiler-independent CMake ([CMake](#)), which is compatible with a variety of compilers, such as GCC/G++ and MS Visual Studio. Fig. 2 shows the main entry page of DiffusionKit, which consists of two core modules: Processing and Visualization. Each function is standalone, and the main features are simply organized by the GUI. Such a design fully facilitates batch processing for large datasets by means of scripting, and it also supports future enrichment of functions of the pipeline.

2.2 Data preprocessing

Before modeling the distribution function within voxels, the DWI images are corrected and aligned for precise modeling. First, for the convenience of processing in DiffusionKit, the raw data are converted from DICOM images to a single 4D NIFTI image (to save storage space, DiffusionKit uses files in zipped format, e.g., *nii.gz*, by default). This step utilizes *dcm2nii*, developed by Dr. Chris Rorden ([MRICron](#)), which is fast and stable and has been well tested within the community ([Jenkinson et al., 2012](#)). Occasionally this program fails to extract the gradient table from the DICOM image series (some MRI scanners or the associated PACS systems use specific techniques to arrange the keyword dictionary of the DICOM), so we have provided a temporary amendment using Matlab code on the FAQ section of the Web site: <http://diffusion.brainnetome.org>. Hence, three files are generated: one is the 4D DWI volume series, and the other two are the b-value table and gradient direction table. These three files are used in several subsequent steps.

The 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 ([Andersson and Sotiropoulos, 2016](#); [Bernstein et al., 2004](#)). Advanced correction mechanism for susceptibility-induced distortion when data acquisition with different phase-encode parameters is becoming increasingly popular. To include the correction method using different phase-encode information, we have exported the functions of *topup*, *applytopup*, *eddy* and *eddy_combine* from FSL ([Andersson and Sotiropoulos, 2016](#); [Andersson et al., 2003](#); [Smith et al., 2004](#)), compiled them on both Linux and Windows platforms, and packed the executable files into the latest version of DiffusionKit. 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, using a strategy similar to the *eddy_correct* in FSL ([Jenkinson et al., 2012](#)). The implementation of this step utilizes the module of NiftyReg, and it harnesses a multi-scale-based block-matching strategy to estimate the location displacements between images ([Ourselin et al., 2001](#)). Subsequently, a transformation model is also applied for the gradient table by the *bnrotate_bvec* function. If the DWI dataset contains multiple reference images (namely b0 images), one can specify which one is the target image during the correction and alignment.

Because the image content outside the brain tissue is irrelevant, the next step strips this part of the image from the brain tissue using a deformable model evolved to locate the brain's surface ([Smith, 2002](#)). Although this step is not compulsive for DiffusionKit, it is always recommended to save computation time on these irrelevant data.

It is necessary to check the results for each step during medical image processing and hence the GUI of DiffusionKit provides a handy check tool in each step (by the checkbox). This design also applies to the sequential modeling and tracking steps.

2.3 Modeling within voxels

To date, there have been a large number of modeling strategies for the distribution function within voxels. Considering the computational burden and the applicability to practical data collections, DiffusionKit adopts two typical modeling methods: one is the popular DTI method, and the other is the HARDI method ([Xie et al., 2015](#)).

Generally, under a diffusion-weighted spin-echo EPI pulse sequence, the propagation of water molecules will be hindered by the white matter microstructures, and the detected DWI attenuated signal $S_i(\mathbf{q})$ is expressed as ([Cheng et al., 2013](#); [Descoteaux et al., 2006](#))

$$S(\mathbf{q}) = S \int_{\mathbb{R}^3} P(\mathbf{R}) \exp(-2\pi i \mathbf{q} \mathbf{R}) d\mathbf{R}, \quad (1)$$

where $P(\mathbf{R})$ is the diffusion probability density function called ensemble average propagator (EAP) to describe the ensemble mean probability in the voxel that the water molecules move

with displacement \mathbf{R} , S_0 is the reference image collected without diffusion gradients, and \mathbf{q} is the wave vector in \mathbf{q} -space, which is dependent on gyromagnetic ratio, the i -th diffusion gradient vector, etc. When we simply assume a Gaussian distribution for the $P(\mathbf{R})$, Eq. 1 can be simplified as (Alexander et al., 2007; Basser et al., 1994),

$$S = S_0 \exp(-b\mathbf{g}^T \mathbf{D} \mathbf{g}_i), \quad (2)$$

where b is the general diffusion weighting factor, usually called the b -value, \mathbf{g}_i is the i -th gradient direction, and the \mathbf{D} is the diffusion tensor characterized by the following positive definite matrix (Alexander et al., 2007; Basser et al., 1994),

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}, \quad (3)$$

Tensor \mathbf{D} is a direction-independent descriptor that is able to characterize the properties of white matter microstructures while avoiding the effects of the gradient directions. From the above tensor \mathbf{D} , its eigenvalues can be readily drawn as $\lambda_1, \lambda_2, \lambda_3$. The details of calculate tensor \mathbf{D} are included in the Supplementary Materials.

This model is useful for extracting a series diffusion index, such as FA, MD and RA, as described by (Alexander et al., 2007)

$$\begin{aligned} \text{FA} &= \frac{\sqrt{3} \left\| \mathbf{D} - \frac{1}{3} \text{Trace}(\mathbf{D}) \mathbf{I} \right\|}{\sqrt{2} \|\mathbf{D}\|} = \sqrt{\frac{3}{2}} \sqrt{\frac{(\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}, \\ \text{MD} &= \bar{\lambda} = \frac{1}{3} \text{Trace}(\mathbf{D}) = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}, \\ \text{RA} &= \sqrt{\frac{(\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2}{3\bar{\lambda}}}, \end{aligned} \quad (4)$$

where λ_i ($i = 1, 2, 3$) indicate the eigenvalues of the diffusion tensor \mathbf{D} and $\bar{\lambda}$ is the mean eigenvalue.

Although, to solve Eq. 2, only 6 noncollinear gradient directions plus one b_0 image are required, usually more than 6 gradient directions are recommended for a more robust solution. Additionally, in the ideal case, the diffusion attenuated signal S_i should be strictly lower than the baseline image S_0 . However, due to various types of noise existing on DWI images, such as the eddy current and head motion, the above case will occasionally be violated, particularly in the cerebrospinal fluid (CSF) or outside the brain tissue; thus, these types of noised-signals will inevitably deteriorate the solution of Eq. 2. Different researchers have devised different methods to deal with this problem, which will be described in the Results and discussion section.

For the HARDI method, DiffusionKit chooses two representative methods (Xie et al., 2015): the decomposition-based SPFI method (Cheng et al., 2010a; Cheng et al., 2010d) and the deconvolution-based CSD method (Tournier et al., 2007). The SPFI estimates both the orientation distribution function (ODF) and ensemble EAP profile by representing the diffusion signal using the SPF basis; this process works well even for voxels with noise and low anisotropy. It also supports a multi-shell DWI data collection scheme.

In SPFI method, which was first proposed by Assemlal et al. (Assemlal et al., 2009), the

diffusion signal attenuation can be represented by the spherical polar Fourier basis functions in Eq. 5

$$E(\mathbf{q}\mathbf{u}) = \frac{S_i(\mathbf{q})}{S_0} = \sum_{n=0}^N \sum_{l=0}^L \sum_{m=-l}^l a_{nlm} G_n(q) Y_l^m(\mathbf{u}), \quad (5)$$

$$G_n(q) = \left[\frac{2n!}{\zeta^{3/2} \Gamma(n+3/2)} \right]^{1/2} \exp\left(-\frac{q^2}{2\zeta}\right) L_n^{\frac{1}{2}}\left(\frac{q^2}{\zeta}\right), \quad (6)$$

where, $\mathbf{q}=\mathbf{q}\mathbf{u}$ and \mathbf{u} is the unit vector, ζ is the scale factor, Γ is the Gamma function, Y is the real spherical harmonic (SH) basis, and L is the generalized Laguerre polynomial ([Assemlal et al., 2009](#); [Cheng et al., 2010a](#); [Cheng et al., 2010d](#)). Furthermore, Cheng and his colleagues proposed an analytical solution for transforming the coefficients of $E(\mathbf{q})$ to the coefficients of the diffusion ODF proposed by [Wedeen et al. \(2005\)](#) and the coefficients of EAP profile represented by the spherical harmonics basis. It was reported that the EAP profile usually had greater ability to characterize the main diffusion directions than ODF ([Cheng et al., 2010d](#)). The implementation of SPFI is provide in the Supplementary Materials.

The CSD method assumes that multi-fiber distribution within voxels can be achieved by convolution between a single fiber response function and the fiber orientation density function (fODF/FOD). Furthermore, it is also assumed that all of the white matter regions of the brain share the same diffusion characteristics, except that different numbers of fibers are oriented with specified angles and weights. Hence, for one voxel, the measured DWI signal can be expressed as the convolution between the FOD and the general response function ([Tournier et al., 2004](#)).

$$S(\theta, \phi) = F(\theta, \phi) \otimes R(\theta), \quad (7)$$

where $R(\theta)$ is the response function, which must be estimated from the well-aligned white matter regions (in practice, judged by the FA value), and $F(\theta, \phi)$ is the fiber orientation density function. When the response function is obtained, similar to the SPFI method described above, representing both $F(\theta, \phi)$ by spherical harmonic basis, we can utilize the deconvolution of $R(\theta)$ from $S(\theta, \phi)$ to estimate the FOD. A reliable solution for Eq. 7 is obtained by including a constraint on the negative values of the FOD ([Tournier et al., 2007](#)).

Because these two methods represent their results by SH basis, DiffusionKit unifies the coefficient alignment, as described in Table 2. This unification facilitates the following fiber tracking and visualization steps.

2.4 Fiber tracking

DiffusionKit currently provides a streamlined fiber tracking method for both DTI and HARDI modeling results. The DTI dataset contains eigenvector information that indicates the main directions within the voxels, resulting in the implementation of streamlined strategy that is straightforward ([Basser et al., 2000](#); [Conturo et al., 1999](#)). For the HARDI dataset, the main directions should first be located across the entire ODF/FOD of each voxel. DiffusionKit applies hierarchical resolution for the ODF/FOD surface representation, enabling rapid, rough localization of the peaks as the initial positions and then precisely searching within the constrained local surface by the gradient descent algorithm. The illustration of this strategy is presented in Fig. 3. In panel A, the red ball indicates one of the local maxima after a global search of the entire ODF/FOD. Because such a search process is only for a rough triangulated mesh of the distribution, it is relatively rapid. In the current design of DiffusionKit, we

perform a discrete search on a coarse mesh (81 directions in a hemisphere from an icosahedral tessellation) to locate possible local maxima. Then, we remove the local maxima with values less than 1/4 of the maximal value among the 81 values. Henceforth, as shown in panel B of Fig. 3, we set the remaining maxima as initial points and perform a gradient descent search to update the maxima precisely along the continuous surface. Such a two-step design effectively avoids falling into nuisance local maxima aroused by noisy signals, while keeping a rapid and effective search.

In seeking good compatibility with the existing toolkits in the literature, DiffusionKit adopted the TrackVis track file format (.trk) ([Wang et al., 2007](#)), whose header was adapted from the widely used NIFTI data format, including the voxel spacing parameters, orientation matrix, x/y/z flip flags, etc. Furthermore, we have enriched the “reserved section” using some pre-computed properties of fiber bundles, the version number of the current track format, the total number of fibers, the mean length/FA/MD of the fiber contained in this file and the whole volumes of the fiber bundles in float type (single precision). Additionally, the attributes (FA/MD and fiber length) of each fiber have also been stored in the extended section. These pre-computed statistical attributes of the fiber bundle, in fact, are by products of DiffusionKit during the fiber tracking process, which is also the case for all of the other implementations in the existing toolkits, so we saved these attributes in the “reserved section” both for visualizing the fiber bundle and for connectivity-based analysis of the brain network. All of the details of the data format for DiffusionKit are also detailed on the DiffusionKit Web site.

2.5 Viewing and visualization

The combination of the analysis steps and visualization steps is a critical feature of DiffusionKit. This combination facilitates both the results checking during processing and the visualization of the final results (e.g., white matter fibers). DiffusionKit is able to visualize several types of images, including 3D/4D NIFTI data sets, region of interest (ROI) overlays, DTI/ODF/FOD glyphs and fiber bundles. There are various configurations for the objects, such as colors, transparency, section transition, and hidden on/off.

The input files for visualization are basically classified into four types -- background, ROI, fiber bundle and diffusion glyph (including diffusion tensor and FOD/ODF) -- and each type of files has its own specific attributes. The background images are usually volume images, such as T1/T2 or single DWI/fMRI volumes, and one of the important roles is to facilitate the location of the anatomical regions of interest, while for the 2D show panel, the software is also able to support multiple background image importation, which is used to compare different background layers. For example, it is able to validate whether the alignment of two images is sufficient. The 3D show panel supports only one active image at a time, and it supports surface/volume rendering, which is implemented by freely moving the intensity and opacity curve in the dedicated control panel. For the ROI images, there are switches to determine whether the specific ROI file will appear in the 2D/3D show panels. Additionally, there are also color and opacity settings for individual ROI files, which are utilized to adjust whether they are overlaid or specifically labeled. For the white matter fiber data, one can choose the rendering types as lines (faster rendering) or tubes (fancy but, in contrast, slower), the color mode (by the fiber directions or mono-color for individual fiber bundle) and the opacity factor for each fiber bundle. For the glyph images, including tensors

and FOD/ODF, it provides a versatile tool to validate whether the modeling results for the individual voxel are correct. One can choose the rendering resolution for each glyph (depending on the computer graphic card) and the coloring mode for the glyph (by directions or by pointing to the ODF origin distance). Specifically, one can overlay the main directions on the individual glyph, and the main directions for each voxel are computed by the *bnSH2Direction* function; please refer to the next section, Miscellaneous tools, and the DiffusionKit Web site for more information.

In the Results and discussion section, some of these features will be demonstrated; please refer to the “-h” argument for help when you execute specific function and the Web site for more detailed descriptions.

2.6 Miscellaneous tools

To establish a complete pipeline for constructing the anatomical brain network, DiffusionKit delivers a series of tools that facilitate image and white matter fiber manipulations. The following content will describe some important functions of DiffusionKit, which have not been adopted in the GUI.

2.6.1 Quality assurance

Quality assurance is a comprehensive and significant issue in the area of diffusion MRI. Many factors and types of artifacts can degrade the quality of diffusion MRI image. Using diffusion MRI data with poor quality can negatively affect the results of studies, consequently leading to wrong conclusion ([Oguz et al., 2014](#)). We provide a preliminary strategy to facilitate quality control. We include the function *bnQA* in DiffusionKit which calculates the mean difference ($S_{diff-mean}$) between the measured DWI signal value and the estimated signal value based on the reconstructed tensor as defined in Eq. 8 and stores the values into a 3D Nifti file. N is the number of gradient directions and $\mathbf{g}(i)$ is i -th gradient direction. s_0 is the signal without diffusion gradient and \mathbf{D} is the diffusion tensor. This utility can help users find out some kinds of artifacts or noise efficiently as shown in Fig. 4.

$$S_{diff-mean} = \frac{\sum_{i=1}^N |s_{measured}(\mathbf{g}(i)) - s_{estimated}(\mathbf{g}(i))|}{N} = \frac{\sum_{i=1}^N |s_{measured}(\mathbf{g}(i)) - s_0 * e^{-\mathbf{B}_i \mathbf{D}}|}{N}, \quad (8)$$

$$\mathbf{B} = b(i) \cdot [g_x g_x(i) \quad 2g_x g_y(i) \quad 2g_x g_z(i) \quad g_y g_y(i) \quad 2g_y g_z(i) \quad g_z g_z(i)], \quad (9)$$

$$\mathbf{D} = [D_{xx} \quad D_{xy} \quad D_{xz} \quad D_{yy} \quad D_{yz} \quad D_{zz}]^T, \quad (10)$$

2.6.2 Image processing

Image registration is a critical step in image processing. DiffusionKit integrates a powerful and elegant toolkit, NiftyReg ([Modat et al., 2010](#); [Ourselin et al., 2001](#); [Rueckert et al., 1999](#)), as the registration module, where it implements a multi-scale-based block-matching strategy to measure the location displacements between images ([Ourselin et al., 2001](#)). It contains several key steps for the image registration process, including the initial rough rigid/affine transformation (the *reg_aladin* function), precise nonlinear registration (the *reg_f3d* function) and the operation of applying a deformation matrix (the *reg_resample*

function). The *bncalc* module implements some basic operations on images, such as the addition, subtraction, multiplication and division operations, and it creates ROIs given the coordinates defining the ROIs. The *bnmerge/bnsplit* pair merges a series of 3D volumes (such as DWI volumes or fMRI volumes) into an integrated 4D image, and vice versa. The *bnroisplit* function breaks the ROI definition file, such as an AAL template file, into a series of ROI files, each of which contains only one ROI, and the name is the ROI label given by the “-l” parameter.

2.6.3 FOD/ODF data processing

The *bnSH2Direction* function is used to compute the possible main directions given a FOD/ODF expression (the coefficients of spherical basis are stored as a NIFTI file, as in Table 2).

2.6.4 Fiber manipulation

Once we obtain the tractography of the desired regions, it is possible to construct an anatomical connectivity network of the brain. The fiber manipulation module is designed to prune/combine the generated white matter bundles by means of and/or/not logical operations. When one specifies a white matter bundle that connects two given ROIs, according to the existing anatomical knowledge, such a track should or should not go through another specified ROI, as illustrated in Fig. 5, and this requirement is able to be implemented easily by the fiber manipulation module, including *bnfiber_end* and *bnfiber_prune*. In Fig. 5, there are three ROIs, labeled A, B and C. If one wants to obtain the pink fiber bundle from the white brain fibers, then he or she can use “and B and C” to prune the original fibers. Similarly, if one wants to obtain the black bundle, then he or she can use “and A and C not B”, and if one wants to obtain all of the fibers, he or she can use “and A” or “or B or C”.

The *bnfiber_map* function generates the fiber density map from the whole tractography ([Calamante et al., 2010](#)).

2.6.5 Anatomical network construction

The pruned fiber bundle is able to define the connectivity between two anatomical regions, and some critical attributes are stored in the reserved section of the .trk file, such as the total number of fibers, mean FA, mean MD, etc. However, DiffusionKit provides a comprehensive function, *bnnetwork*, to construct the connectivity network easily from the track results.

3. Results and discussion

The DiffusionKit package is implemented in C/C++ and Qt/VTK, is freely available at <http://diffusion.brainnetome.org> and <https://www.nitrc.org/projects/diffusionkit>. Table 3 lists the main functions of DiffusionKit, and the full manual and associated tutorials are available from the website: <http://diffusion.brainnetome.org>. The following results are obtained using DiffusionKit.

3.1 Validations of diffusion tensor computing

The diffusion tensor and its associate attributes (e.g., FA) calculations are basic functions of DiffusionKit. For this part, we adopted one subject DWI dataset acquired from a GE Discovery 750 3T MRI scanner at the University of Electronic Science and Technology of China. These data were also used in our previous study ([Xie et al., 2015](#)); for a self-contained description, the main parameters are repeated here: field-of-view (FOV) = 256 * 256 mm², matrix size = 128 * 128, slice = 75, voxel size = 2 * 2 * 2 mm³ and without a slice gap. The current work used the dataset with b-value = 1000 s/mm² on 64 noncollinear diffusion gradient directions. For impartial validation, the raw data, which are available through the DiffusionKit Web site, were pre-processed using existing software, dcm2nii and FSL 5.0.6 (eddy_correct + bet2). Then, the processed data were fed into the software to generate the FA results.

It should be noted that we did not conduct statistical analysis of a cohort of subjects because the comparisons of the voxel-wise attributes were conducted in the native DWI space, and the inter-object alignment required by the group comparisons would inevitably have introduced unexpected errors that would have deteriorated the nature of the comparisons.

Herein, the comparisons are conducted between the DiffusionKit (abbr. DK) and two other well-known packages: FSL/FDT (abbr. FSL) ([Jenkinson et al., 2012](#)) and Diffusion Toolkit (abbr. DT) ([Wang et al., 2007](#)). Fig. 6 shows the FA discrepancy separately for DK vs. FSL (panel A), DT vs. FSL (panel B) and DK vs. DT. Figs. 6-8 are all shown by MRICron ([MRICron](#)). The non-colored regions indicate that the differences between the two software programs are negligible (the absolute value is less than 1.0e-6). Fig. 6 shows that, in the white matter regions, there is nearly no significant difference between DK and FSL, and the differences in panels B and C also approach 0. For a zoom-in check of these difference, in Fig. 7, the color window is set as [0.0005, 0.025], and this setting removes most of the error regions in Fig. 6, indicating that the errors among these three groups of comparisons are not confined to the white matter regions. Similarly, in Fig. 8, the first eigenvector of the tensor is compared for the three groups, and it also shows the identical error patterns as in Figs. 6 and 7. It should be emphasized that, compared to the FSL, DK and DT have similar error patterns, which remain in the non-white matter regions. Actually, in these non-white matter regions, such as the CSF areas, the collected DWI signal usually are not stable, resulting S_i (the measured signal value of the i -th diffusion-weighted volume) $> S_0$ (the measured signal value of the non-diffusion-weighted volume) for those voxels. Although some methods have been proposed to handle with suchlike ill-posed equations, different researchers have not reached a consensus yet in the literature ([Veraart et al., 2013](#)), so in current stage, DiffusionKit does not use any weights to solve the problems.

3.2 Validations of the FOD computing

Because the analytical SPFI method was proposed by our group ([Cheng et al., 2010a](#); [Cheng et al., 2010d](#)), and it has been adopted in other papers ([Cheng et al., 2013](#); [Xie et al., 2015](#)), it is not necessary to compare its implementation to others. Hence, we only conducted the comparisons for the FOD computing between the DiffusionKit and MRtrix ([Tournier et al., 2012](#)). Fig. 9 shows the FOD reconstruction results based on the DWI data described above,

with $b=2000$ s/mm², which was the recommended protocol in Xie *et al* (2015), and the same fiber crossing region, the centrum semiovale. The two columns of Fig. 9 separately indicate the left and right centrum semiovale, with row A from DiffusionKit and row B from the MRtrix package, while row C is voxel-wise absolute angle difference of the main directions between rows A and B. Visual inspection of rows A and B did not show significant difference; furthermore, row C showed that the angular difference between rows A and C were no larger than 8 degrees, which is a reasonable error because it is in agreement with the mesh precision for searching in Fig. 3.

3.3 Fiber tracking of different methods in DiffusionKit

This section shows several modeling and fiber tracking results implemented by DiffusionKit. The dataset with 2000 s/mm² was from the “Fiber Cup”, the intention of which is to evaluate the performance of different modeling and tracking approaches (Fillard *et al.*, 2011). Figs. 10 and 11 show the modeling and tractography results, respectively, by DTI/SPFI/CSD. The DTI method failed to distinguish the crossing fibers (see Fig. 11D), while the SPFI and CSD methods were successful, in agreement with what has been reported in the literature. The details and source script for how to estimate the tensor/ODF/FOD and track the fibers are included in the Supplementary Materials and are also available on the DiffusionKit Web site.

Additionally, using the ROIs defined in Wakana *et al*'s paper (2007), we further validated the tractography module of DiffusionKit. Fig. 12 shows the tractography results using the same data as in Figs. 6-8 ($b = 1000$ s/mm², DTI modeling) from the part of the corticospinal tract (panel I) and the cingulate gyrus (panel II), which are in agreement with the Figs. 4 and 2 in (Wakana *et al.*, 2007), respectively.

3.4 Construction of an anatomical network

Here, we demonstrate how DiffusionKit constructs an anatomical brain network based on DWI image series. As a short example, the diffusion data of one subject from HCP Q1 release (#100307) (Van Essen *et al.*, 2013) were adopted, containing multi-shell collections of $b=1000$, 2000 and 3000 s/mm², and this was also to validate the ability of SPFI on the reconstruction based on multi-shell data. The whole brain fiber tracks were pruned by ROI pairs from the AAL template (only the cerebral part), which defined the start and end regions connected by the resulting fibers. Fig. 13 shows the connectivity network derived from the 90 ROIs weighted by the number of fibers (indicated by the color bars), which demonstrated strong similarity to the results of Li *et al* (see Fig. 2 in (Li *et al.*, 2009)), which used an averaged binarized network. The details and source script concerning how to construct the network are described in the Tutorial section on the website of DiffusionKit.

4. Conclusions

In this work, we developed a light, one-stop package called DiffusionKit for diffusion MRI data analysis, which included all of the required steps from the original DICOM images to brain anatomical network construction and user-friendly operation and visualization. Along

the entire pipeline, various types of visualization schemes dedicated to different results were also provided.

In contrast with the existing toolkits in the literature, DiffusionKit has comparable features and merits. In Table 2, popular tools in the community that have similar functions are listed and compared with DiffusionKit. This comparison is specifically focused on brain anatomical network construction and image/fiber visualization, which are the main features of DiffusionKit. Table 2 demonstrates that DiffusionKit has a complete pipeline for dMRI data analysis and visualization, with a small installation size and cross-platform support.

In summary, DiffusionKit provides an all-in-one solution for researchers to perform dMRI studies, especially for those who are not acquainted with corresponding algorithms and proficient in programming. For both expert and non-expert users, the installation is rapid and convenient. The self-contained installation package is approximately 50 M (<15 M for MS Windows). The users do not require any other software to perform a complete dMRI data analysis pipeline, including data format conversion, preprocessing, local modeling and reconstruction, fiber tracking, fiber statistics and visualization. DiffusionKit was designed for interaction via a GUI or command-line window. The GUI interface delivers most of the basic tools and parameters for configuration necessary to run the whole pipeline. For advanced users who are able to use scripts, command-line calling can be used to perform batch computing for large datasets. The toolkit can serve as a standalone image viewer for T1/dMRI/DTI/HARDI images as well.

Acknowledgement

This work was supported in part by the National Key Basic Research and Development Program (973) (Grant 2011CB707800), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDB02030300), the Natural Science Foundation of China (Grants 81000634 and 81270020) and the National High-tech R&D Program (863) (Grant 2015AA020513). The authors thank Dr. Zhengyi Yang, Dr. Jing Sui and Dr. Shan Yu for their assistance in language editing and for the fruitful discussions. The corresponding author also thanks Dr. Jacques-Donald Tournier for sharing the MATLAB code of the CSD method, which inspired rapid C/C++ implementation. The data (diffusion data of #100307 from Q1 release) in the Results section were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

Figure Legends

Figure 1. The overall design framework and the main modules of DiffusionKit.

Figure 2. The main interface of DiffusionKit, which has two core modules shown in the top-right panel: `Processing` and `Visualization`.

Figure 3. Illustrations of how to search for the main directions from the FOD/ODF mesh. Panel A is a sparse mesh (81 directions in a hemisphere from an icosahedral tessellation) for rough localization, and Panel B is for a precise search constrained in a local area by the gradient descent algorithm.

Figure 4. The example of bnQA for two subjects. The “DWI” column is the original DWI image and the “QA” column is $S_{diff-mean}$ for Subject1 and Subject2. The green arrows indicate the regions affected by artifact or noise and these regions cannot be detected when you checking the original DWI images visually.

Figure 5. Illustrations of how to extract a specific fiber bundles from entire brain tractography, which will utilize logical operations by `bnfiber_prune` function, including “and”, “or” and “not” operations.

Figure 6. Comparisons of FA results among three toolkits: DiffusionKit ([Leemans et al.](#)), FSL ([Jenkinson et al., 2012](#)) and Diffusion Toolkit (`DTITool`) ([Wang et al., 2007](#)). Each panel was generated by the MRICron ([MRICron](#)), and the absolute errors less than $1.0e-6$ were assumed to be not colored. Panels A, B and C separately show pair-wise error maps.

Figure 7. Similar results as Fig. 6, except that the errors less than 0.0005 are not shown.

Figure 8. Comparisons of main direction results among the three toolkits: DiffusionKit ([Leemans et al.](#)), FSL and Diffusion Toolkit (`DTITool`). The display pattern is exactly the same as in Fig. 6.

Figure 9. Validations for the FOD implementation of DiffusionKit. The left and right centrum semiovalae are used to reconstruct the FOD separately by DiffusionKit (Row A) and MRtrix ([Tournier et al., 2012](#)) (Row B). Row C shows the total angle errors for up to 3 main directions for each voxel.

Figure 10. Validations for the voxel-wise distribution reconstructions of DiffusionKit for the “Fiber Cup” phantom. The first row (A-C) shows the overall views of the DTI tensor (panel A), SPFI ODF (panel B) and CSD FOD (panel C). The second row (D-F) shows the zoom-in views of the first row.

Figure 11. Illustrations for the tractography results based on the voxel-wise reconstructions in Fig 10. The first row (A-C) shows the global tractography results and the last row shows the crossing fiber bundles constrained by the ROI pairs, red balls and green balls, in the middle row (panel G). The tract based on DTI fails to show the two branches of the crossing region (panel D).

Figure 12. Validations for the tractography results. Panel I and panel II show two fiber bundles separately constrained by ROI pairs (two groups of A and B, corresponding to Fig. 4 and Fig. 2, respectively, in [Wakana et al., 2007](#)), and both the ROI definitions and fiber tract are extremely close to the paper of Wakana *et al* ([2007](#)).

Figure 13. An anatomical connectivity network constructed by DiffusionKit. The color bar indicates the numbers of fibers connecting the ROI pairs (AAL template) separately along x- and y-axis labels.

References

- Alexander AL, Lee JE, Lazar M, Field AS. Diffusion tensor imaging of the brain. *Neurotherapeutics : the journal of the American Society for Experimental NeuroTherapeutics*, 2007; 4: 316-29.
- Andersson JL, Sotiropoulos SN. An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging. *NeuroImage*, 2016; 125: 1063-78.
- Andersson JLR, Skare S, Ashburner J. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. *NeuroImage*, 2003; 20: 870-88.
- Assaf Y, Pasternak O. Diffusion tensor imaging (DTI)-based white matter mapping in brain research: a review. *Journal of molecular neuroscience : MN*, 2008; 34: 51-61.
- Assemlal HE, Tschumperle D, Brun L. Efficient and robust computation of PDF features from diffusion MR signal. *Medical image analysis*, 2009; 13: 715-29.
- Basser PJ, Mattiello J, LeBihan D. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 1994; 66: 259-67.
- Basser PJ, Pajevic S, Pierpaoli C, Duda J, Aldroubi A. In vivo fiber tractography using DT-MRI data. *Magnetic resonance in medicine*, 2000; 44: 625-32.
- Bernstein MA, King KF, Zhou ZJ. *Handbook of MRI pulse sequences*. Academic Press: Amsterdam ; Boston, 2004.
- Calamante F, Tournier JD, Jackson GD, Connelly A. Track-density imaging (TDI): super-resolution white matter imaging using whole-brain track-density mapping. *NeuroImage*, 2010; 53: 1233-43.
- Chapman S. *MATLAB programming for engineers*. Nelson Education, 2015.
- Cheng J, Ghosh A, Deriche R, Jiang T. Model-free, regularized, fast, and robust analytical orientation distribution function estimation. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2010a; 13: 648-56.
- Cheng J, Ghosh A, Jiang T, Deriche R. Model-free and analytical EAP reconstruction via spherical polar Fourier diffusion MRI. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2010d; 13: 590-7.
- Cheng J, Jiang T, Deriche R, Shen D, Yap PT. Regularized spherical polar fourier diffusion MRI with optimal dictionary learning. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2013; 16: 639-46.
- CMake. <http://www.cmake.org>.
- Conturo TE, Lori NF, Cull TS, Akbudak E, Snyder AZ, Shimony JS, McKinstry RC, Burton H, Raichle ME. Tracking neuronal fiber pathways in the living human brain. *Proceedings of the National Academy of Sciences of the United States of America*, 1999; 96: 10422-7.
- Cook PA, Bai Y, Nedjati-Gilani S, Seunarine KK, Hall MG, Parker GJ, Alexander DC. Camino: Open-Source Diffusion-MRI Reconstruction and Processing. *Proc Intl Soc*

- Mag Reson Med: Seattle, WA, USA, 2006: 2759.
- Cui Z, Zhong S, Xu P, He Y, Gong G. PANDA: a pipeline toolbox for analyzing brain diffusion images. *Frontiers in human neuroscience*, 2013; 7: 42.
- Daducci A, Gerhard S, Griffa A, Lemkaddem A, Cammoun L, Gigandet X, Meuli R, Hagmann P, Thiran JP. The connectome mapper: an open-source processing pipeline to map connectomes with MRI. *PLoS one*, 2012; 7: e48121.
- Descoteaux M, Angelino E, Fitzgibbons S, Deriche R. Apparent diffusion coefficients from high angular resolution diffusion imaging: estimation and applications. *Magnetic resonance in medicine*, 2006; 56: 395-410.
- DTITool. <http://bmia.bmt.tue.nl/software/viste>.
- Fillard P, Descoteaux M, Goh A, Gouttard S, Jeurissen B, Malcolm J, Ramirez-Manzanares A, Reisert M, Sakaie K, Tensaouti F, Yo T, Mangin JF, Poupon C. Quantitative evaluation of 10 tractography algorithms on a realistic diffusion MR phantom. *NeuroImage*, 2011; 56: 220-34.
- FSL. <http://fsl.fmrib.ox.ac.uk>.
- Garyfallidis E, Brett M, Amirbekian B, Rokem A, van der Walt S, Descoteaux M, Nimmo-Smith I, Dipy C. Dipy, a library for the analysis of diffusion MRI data. *Frontiers in neuroinformatics*, 2014; 8: 8.
- Jenkinson M, Beckmann CF, Behrens TE, Woolrich MW, Smith SM. *Fsl. NeuroImage*, 2012; 62: 782-90.
- Jiang HY, van Zijl PCM, Kim J, Pearlson GD, Mori S. DtiStudio: Resource program for diffusion tensor computation and fiber bundle tracking. *Comput Meth Prog Bio*, 2006; 81: 106-16.
- Jiang T. Brainnetome: a new -ome to understand the brain and its disorders. *NeuroImage*, 2013; 80: 263-72.
- Le Bihan D, Mangin JF, Poupon C, Clark CA, Pappata S, Molko N, Chabriat H. Diffusion tensor imaging: concepts and applications. *Journal of magnetic resonance imaging : JMRI*, 2001; 13: 534-46.
- Le Bihan D, Poupon C, Amadon A, Lethimonnier F. Artifacts and pitfalls in diffusion MRI. *Journal of magnetic resonance imaging : JMRI*, 2006; 24: 478-88.
- Leemans A, Jeurissen B, Sijbers J, Jones D. ExploreDTI: a graphical toolbox for processing, analyzing, and visualizing diffusion MR data. *Proc Intl Soc Mag Reson Med*, 2009: 3537.
- Li Y, Liu Y, Li J, Qin W, Li K, Yu C, Jiang T. Brain anatomical network and intelligence. *PLoS computational biology*, 2009; 5: e1000395.
- MIPAV. <http://mipav.cit.nih.gov>.
- Modat M, Ridgway GR, Taylor ZA, Lehmann M, Barnes J, Hawkes DJ, Fox NC, Ourselin S. Fast free-form deformation using graphics processing units. *Comput Meth Prog Bio*, 2010; 98: 278-84.
- Mori S, van Zijl PC. Fiber tracking: principles and strategies - a technical review. *NMR in biomedicine*, 2002; 15: 468-80.
- MRICron. <http://www.mccauslandcenter.sc.edu/mricron>.
- MRtrix. <http://www.mrtrix.org/>.
- Oguz I, Farzinfar M, Matsui J, Budin F, Liu Z, Gerig G, Johnson HJ, Styner M. DTIPrep:

- quality control of diffusion-weighted images. *Frontiers in neuroinformatics*, 2014; 8: 4.
- Ourselin S, Roche A, Subsol G, Pennec X, Ayache N. Reconstructing a 3D structure from serial histological sections. *Image Vision Comput*, 2001; 19: 25-31.
- Rueckert D, Sonoda LI, Hayes C, Hill DL, Leach MO, Hawkes DJ. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE transactions on medical imaging*, 1999; 18: 712-21.
- Smith SM. Fast robust automated brain extraction. *Human brain mapping*, 2002; 17: 143-55.
- Smith SM, Jenkinson M, Johansen-Berg H, Rueckert D, Nichols TE, Mackay CE, Watkins KE, Ciccarelli O, Cader MZ, Matthews PM, Behrens TE. Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data. *NeuroImage*, 2006; 31: 1487-505.
- Smith SM, Jenkinson M, Woolrich MW, Beckmann CF, Behrens TEJ, Johansen-Berg H, Bannister PR, De Luca M, Drobnjak I, Flitney DE, Niazy RK, Saunders J, Vickers J, Zhang YY, De Stefano N, Brady JM, Matthews PM. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 2004; 23: S208-S19.
- Sotiropoulos SN, Jbabdi S, Xu J, Andersson JL, Moeller S, Auerbach EJ, Glasser MF, Hernandez M, Sapiro G, Jenkinson M, Feinberg DA, Yacoub E, Lenglet C, Van Essen DC, Ugurbil K, Behrens TE, Consortium WU-MH. Advances in diffusion MRI acquisition and processing in the Human Connectome Project. *NeuroImage*, 2013; 80: 125-43.
- SPM. <http://www.fil.ion.ucl.ac.uk/spm>.
- Sporns O. The human connectome: a complex network. *Annals of the New York Academy of Sciences*, 2011; 1224: 109-25.
- Sporns O, Tononi G, Kotter R. The human connectome: A structural description of the human brain. *PLoS computational biology*, 2005; 1: e42.
- Tournier JD, Calamante F, Connelly A. MRtrix: Diffusion tractography in crossing fiber regions. *Int J Imag Syst Tech*, 2012; 22: 53-66.
- Tournier JD, Calamante F, Connelly A. Robust determination of the fibre orientation distribution in diffusion MRI: non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 2007; 35: 1459-72.
- Tournier JD, Calamante F, Gadian DG, Connelly A. Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*, 2004; 23: 1176-85.
- Toussaint N, Souplet J, Fillard P. MedINRIA: DT-MRI processing and visualization software. *MICCAI Workshop on Interaction in Medical Image Analysis and Visualization*, 2007; 7.
- TrackVis. <http://www.trackvis.org>.
- Van Essen DC, Smith SM, Barch DM, Behrens TE, Yacoub E, Ugurbil K, Consortium WU-MH. The WU-Minn Human Connectome Project: an overview. *NeuroImage*, 2013; 80: 62-79.
- Veraart J, Sijbers J, Sunaert S, Leemans A, Jeurissen B. Weighted linear least squares

- estimation of diffusion MRI parameters: strengths, limitations, and pitfalls. *NeuroImage*, 2013; 81: 335-46.
- VTK. <http://www.vtk.org>.
- Wakana S, Caprihan A, Panzenboeck MM, Fallon JH, Perry M, Gollub RL, Hua K, Zhang J, Jiang H, Dubey P, Blitz A, van Zijl P, Mori S. Reproducibility of quantitative tractography methods applied to cerebral white matter. *NeuroImage*, 2007; 36: 630-44.
- Wang R, Benner T, Sorensen AG, Wedeen VJ. Diffusion Toolkit: a software package for diffusion imaging data processing and tractography. *Proc Int Soc Mag Reson Med*, 2007; 15: 3720.
- Wedeen VJ, Hagmann P, Tseng WY, Reese TG, Weisskoff RM. Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging. *Magnetic resonance in medicine*, 2005; 54: 1377-86.
- Xie S, Zuo N, Shang L, Song M, Fan L, Jiang T. How does B-value affect HARDI reconstruction using clinical diffusion MRI data? *PloS one*, 2015; 10: e0120773.
- Yeh FC, Verstynen TD, Wang YB, Fernandez-Miranda JC, Tseng WYI. Deterministic Diffusion Fiber Tracking Improved by Quantitative Anisotropy. *PloS one*, 2013; 8.
- Zuo N, Cheng J, Jiang T. Diffusion magnetic resonance imaging for Brainnetome: a critical review. *Neuroscience bulletin*, 2012; 28: 375-88.

Table1

Table 1. Comparisons between DiffusionKit and other toolkits freely available on the internet. The \checkmark and \times indicate “yes” and “no” separately.

Package	DiffusionKit	FSL (Jenkinson et al., 2012)	CMTK (Daducci et al., 2012)	PANDA (Cui et al., 2013)	MRtrix (Tournier et al., 2012)	MriStudio (Jiang et al., 2006)	TrackVis (Wang et al., 2007)	Camino (Cook et al., 2006)	DSI Studio (Yeh et al., 2013)	ExploreDTI (Leemans et al., 2009)	MedInria (Toussaint et al., 2007)	DiPy (Garyfallidis et al., 2014)
Programming language	C/C++	C/C++	Python	MATLAB	C/C++	C/C++	C/C++	Java	C/C++	MATLAB	C/C++	Python
Cross platform (Linux/Win)*	\checkmark	\times	\checkmark	\checkmark	\times	\times	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark
Installation size (MBytes)**	15-50	>300	15	10	1.5	1.8	28	15	54	159	47	5
Standalone	\checkmark	\checkmark	\times	\times	\checkmark	\checkmark	\checkmark	\checkmark	\times	\times	\checkmark	\times
GUI	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark	\times	\checkmark	\checkmark	\checkmark	\times
Command-line	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark
Preprocessing	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	\times	\checkmark	\times	\checkmark	\checkmark	\checkmark
Modeling												
DTI	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
HARDI	\checkmark	\checkmark	\checkmark	\times	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark
Fiber tracking	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
Fiber statistics	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
Visualization												
Slice/Volume	\checkmark	\checkmark	\times	\times	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark
Tensor/ODF/FOD	\checkmark	\times	\checkmark	\times	\checkmark (2D)	\times	\times	\times	\checkmark	\checkmark	\times	\checkmark
Fiber	\checkmark	\times	\checkmark	\times	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark	\checkmark	\checkmark

* indicates that it is able to support at least two operating systems, such as Linux and MS Windows

**indicates a rough size that usually varies across different operating systems, so the installation size in this row is only for reference; it should be noted that some tools, such as FSL and MedInria, integrate other functions than those for dMRI processing. Moreover, some toolkits, such as CMTK and PANDA, only provide script pipeline which requires 3rd party software (FSL, MRtrix, etc) for computing.

Table 2. The coefficient alignment of the ODF/FOD estimation results on the SH basis.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	...
$Y(0,0)$	$\text{Re}Y\{(2,2)\}$	$\text{Re}Y\{(2,1)\}$	$Y(2,0)$	$\text{Im}Y\{(2,1)\}$	$\text{Im}Y\{(2,2)\}$	$\text{Re}Y\{(4,4)\}$	$\text{Re}Y\{(4,3)\}$...

Table 3. List of main functions of the DiffusionKit.

Functions	Called Names	Description (“-h” for more information)
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 the 4D image along the 4th dimension and the opposite side
Modeling	bndti_estimate	Estimate tensor model, output FA, MD, tensor, etc.
	bnhardi_ODF_estimate	Estimate ODF by the SPFI method (Assemlal et al., 2009 ; Cheng et al., 2010d)
	bnhardi_FOD_estimate	Estimate FOD by the CSD method (Tournier et al., 2007)
Tracking	bndti_tracking	Track white matter fiber based on the tensor model
	bnhardi_tracking	Track white matter fiber based on ODF/FOD
Visualization	bnviewer	Visualize various data, e.g., image/ODF/FOD/ROI/fiber
Tools	bncalc/bnroisplit	Calculation, e.g., add/subtract/multiply/divide and ROI generation
	bninfo	Show the head information of DICOM and NIFTI files
	reg_aladin/reg_f3d	Inter/intra-image registration across multiple modalities (Modat et al., 2010 ; Ourselin et al., 2001)
	reg_resample/reg_transform	Resample and apply transformation matrix (Modat et al., 2010 ; Ourselin et al., 2001)
	bnfiber_end/bnfiber_prune	Prune fiber bundle, logical and/or/not based on ROIs, to generate the connectivity bundle for specified ROIs
	bnfiber_stats	Export attributes of fiber bundles, e.g., fiber number, length, FA, etc.
	bnfiber_map	Generate the fiber density going through each voxel (Calamante et al., 2010)
	bnnetwork	A unified function to construct the anatomical network for given ROIs from the whole brain tractography

Figure1
[Click here to download high resolution image](#)

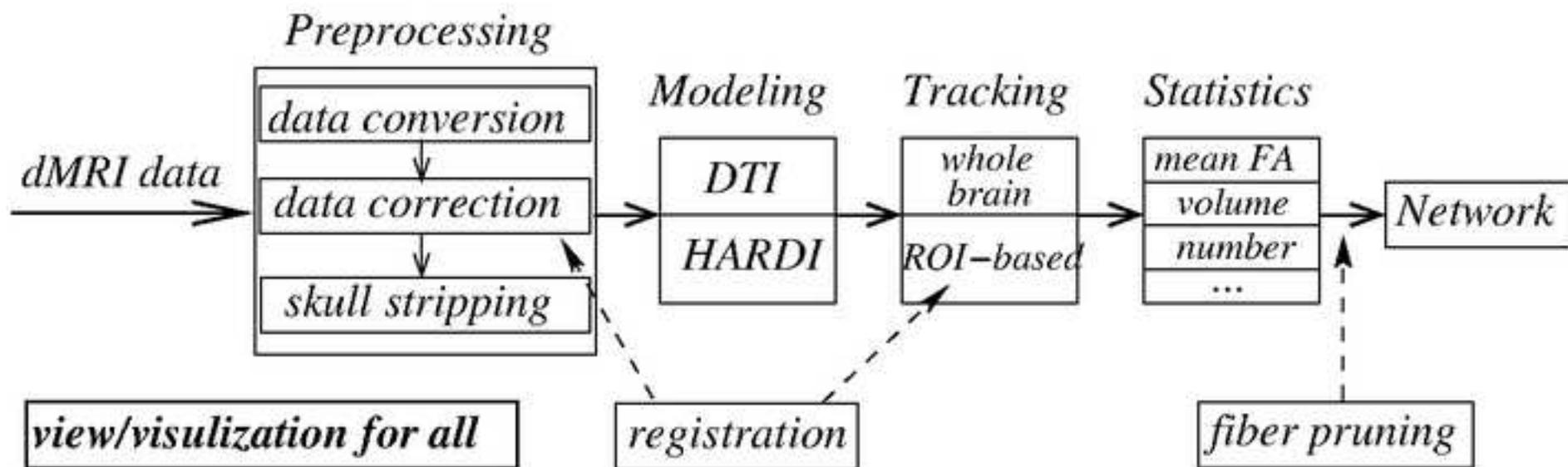


Figure2

[Click here to download high resolution image](#)

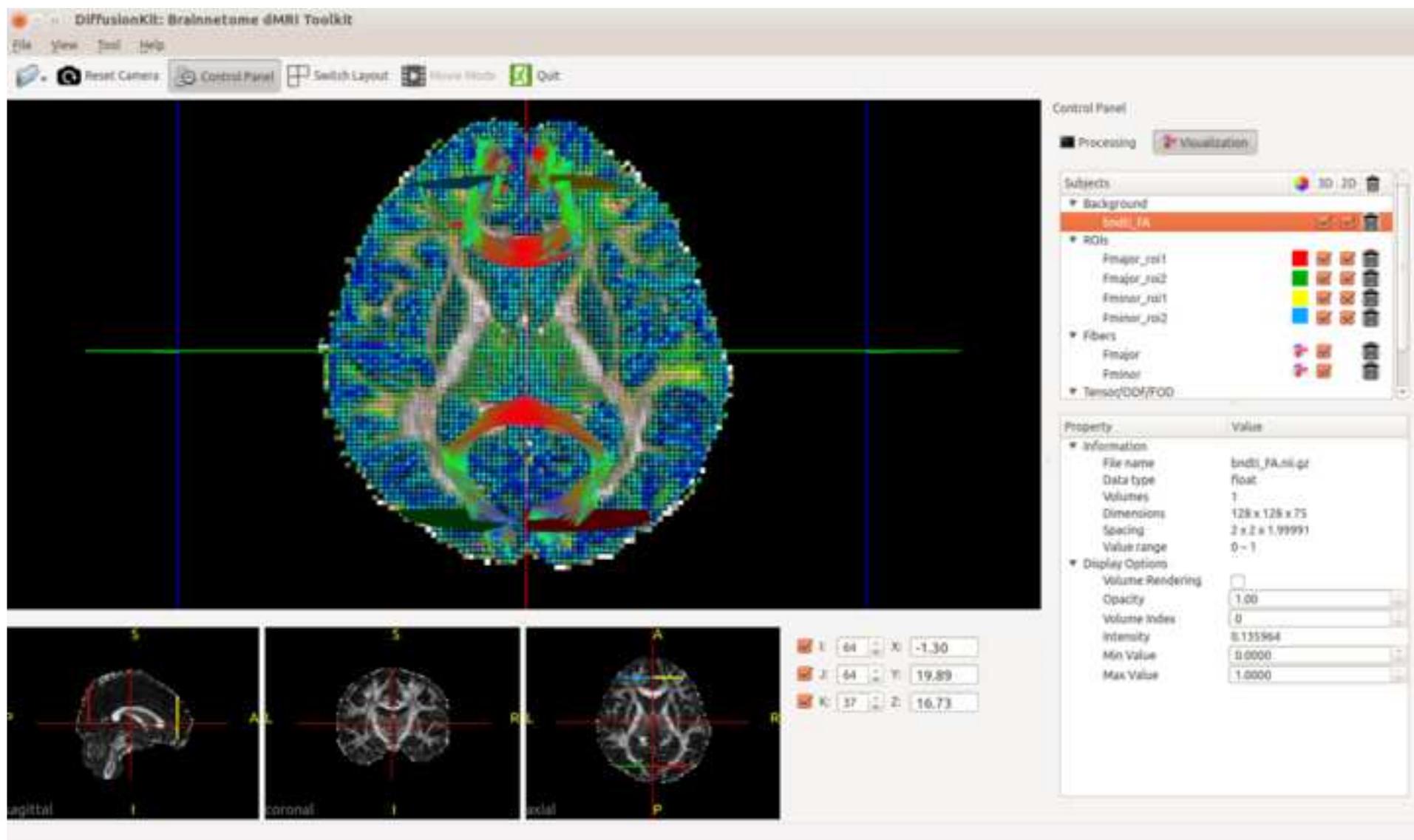


Figure3
[Click here to download high resolution image](#)

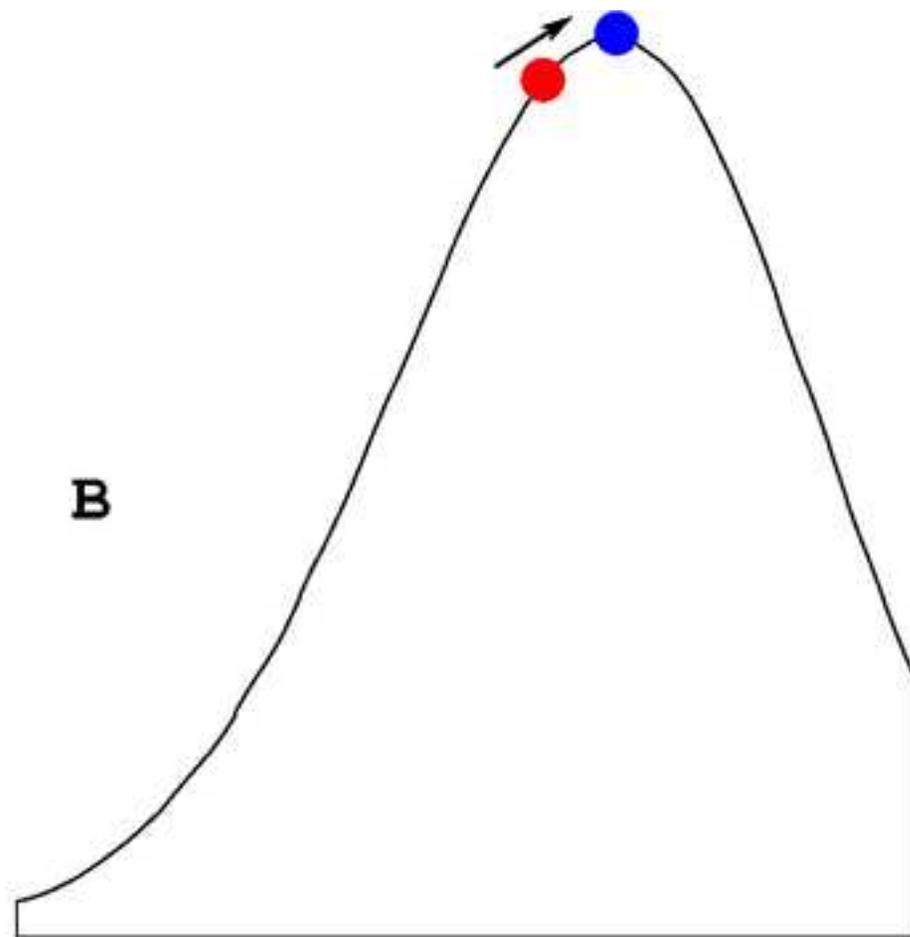
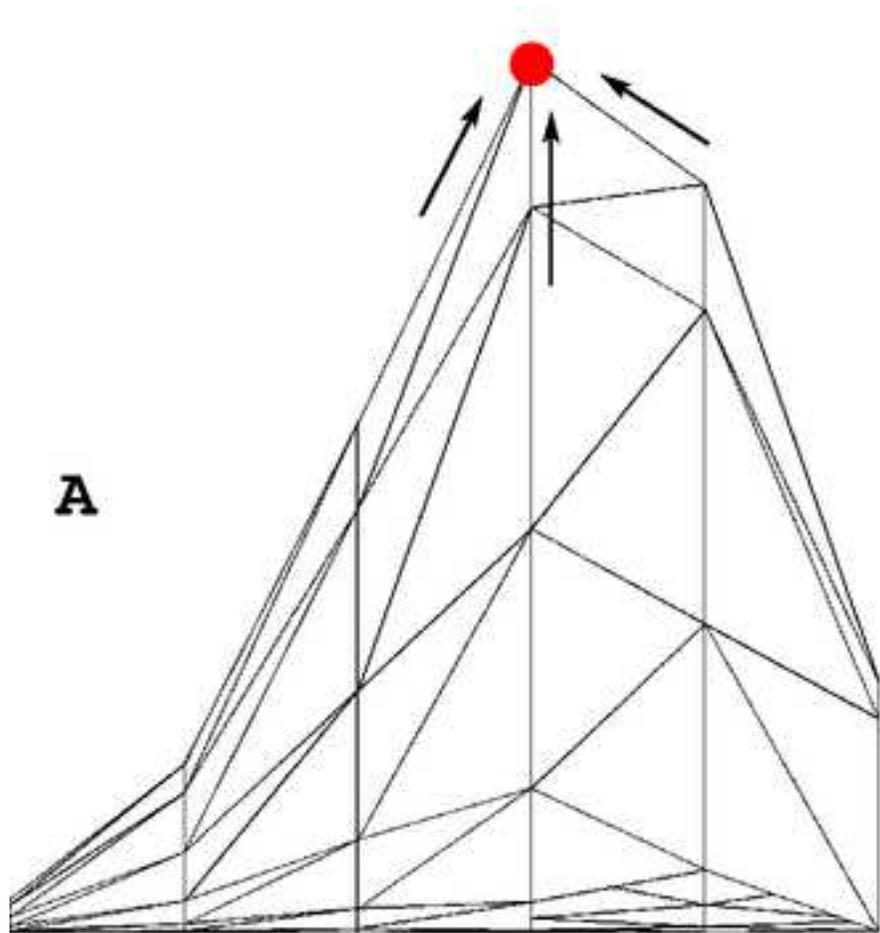


Figure4
[Click here to download high resolution image](#)

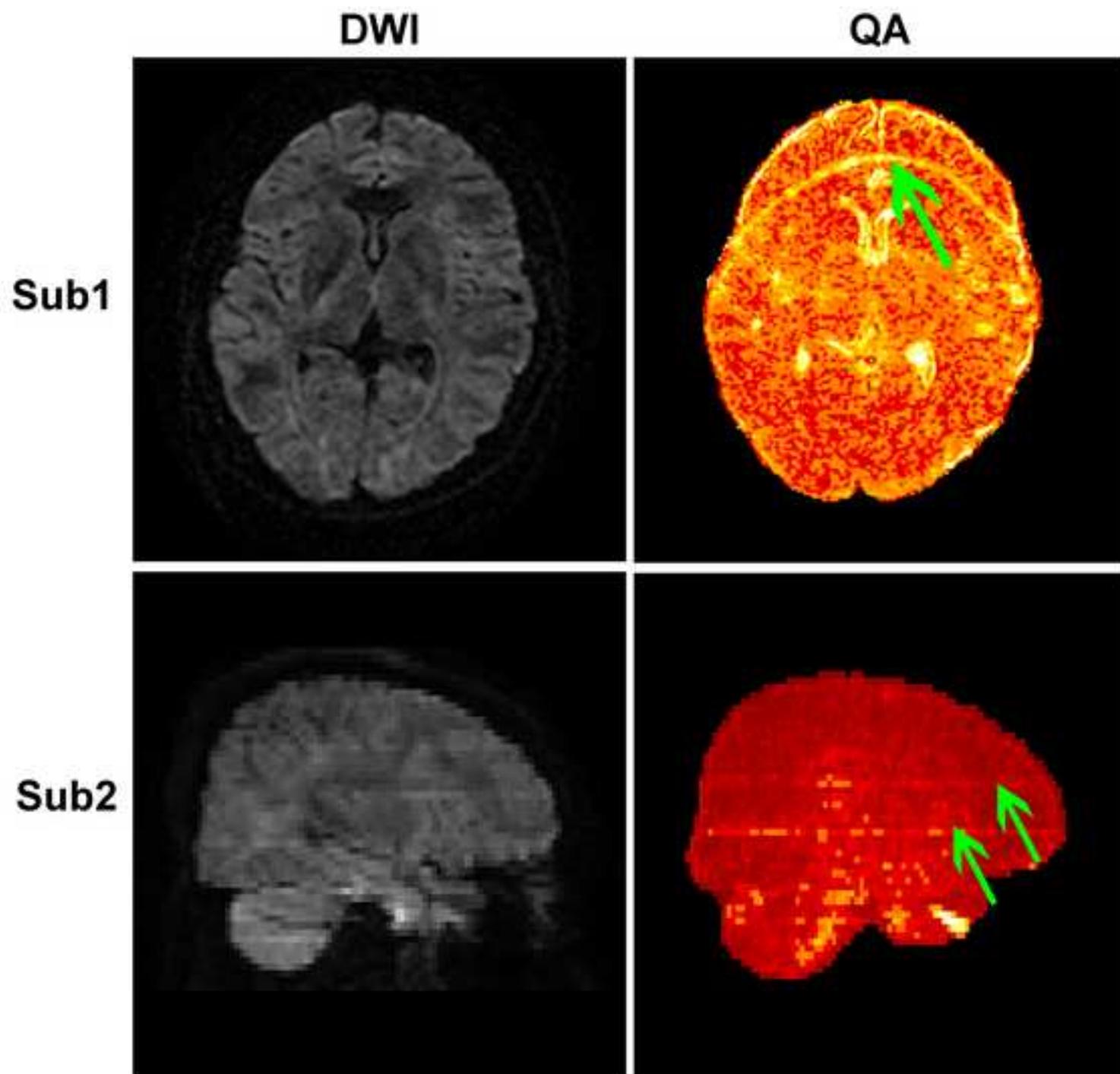


Figure5
[Click here to download high resolution image](#)

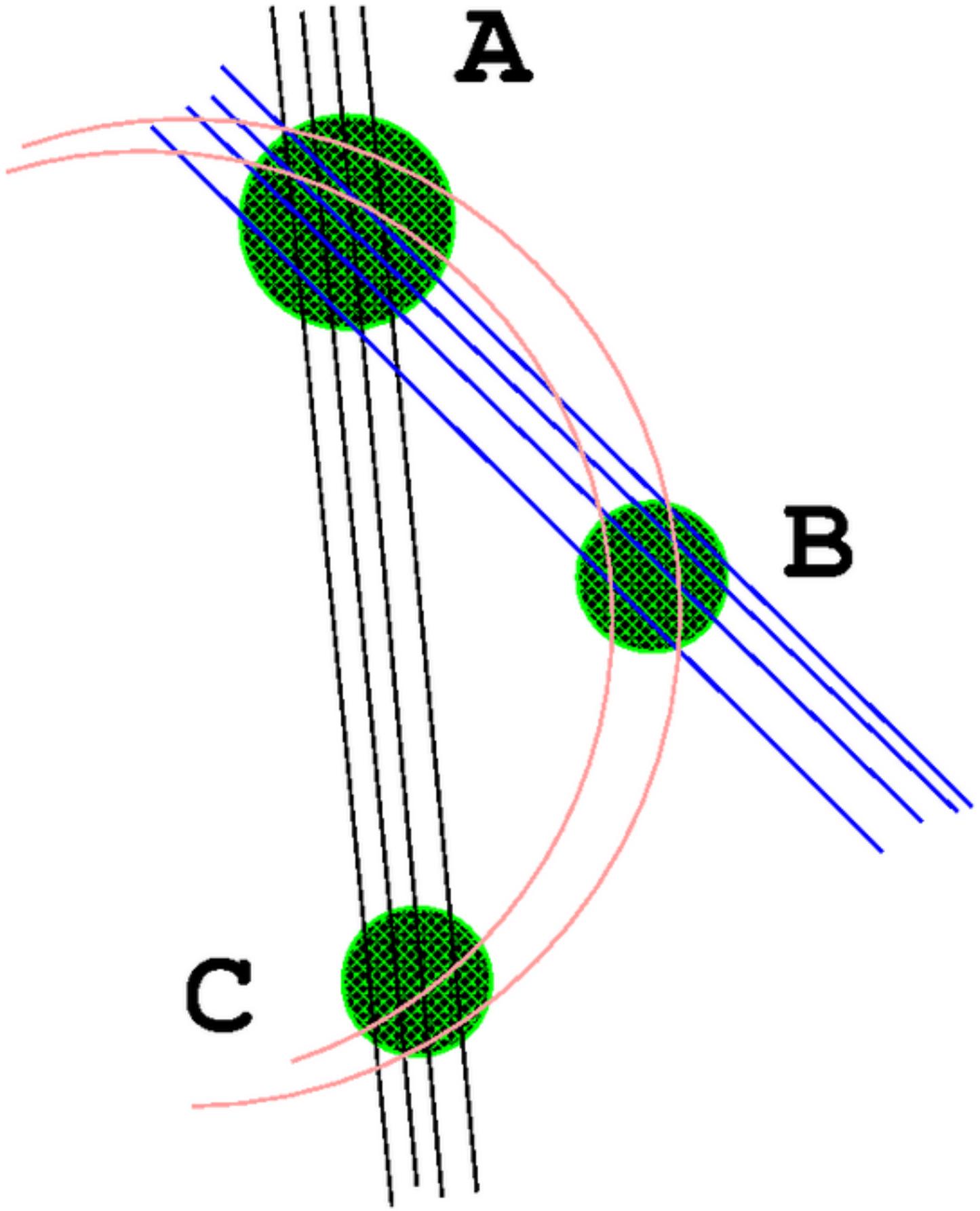


Figure6
[Click here to download high resolution image](#)

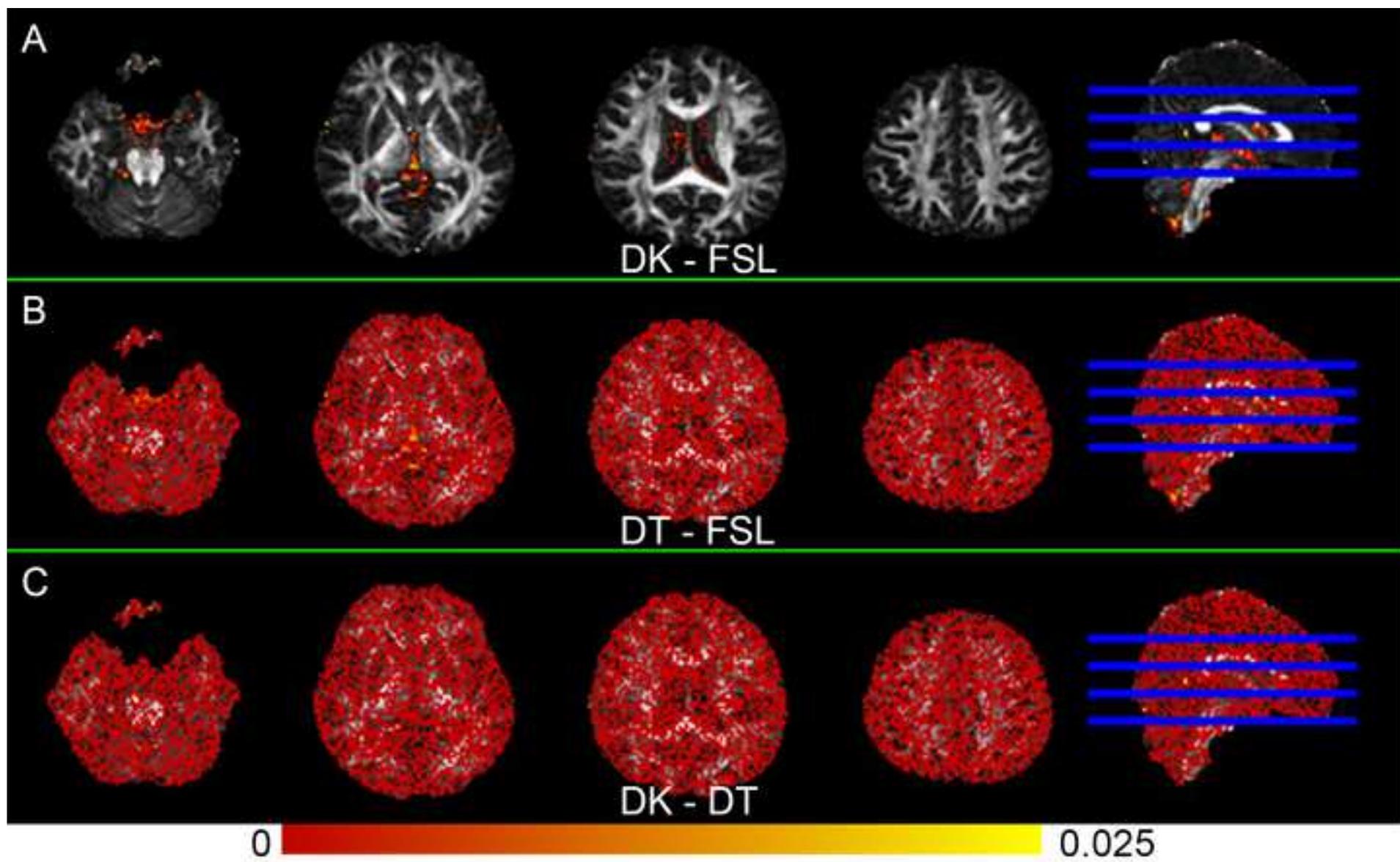


Figure7
[Click here to download high resolution image](#)

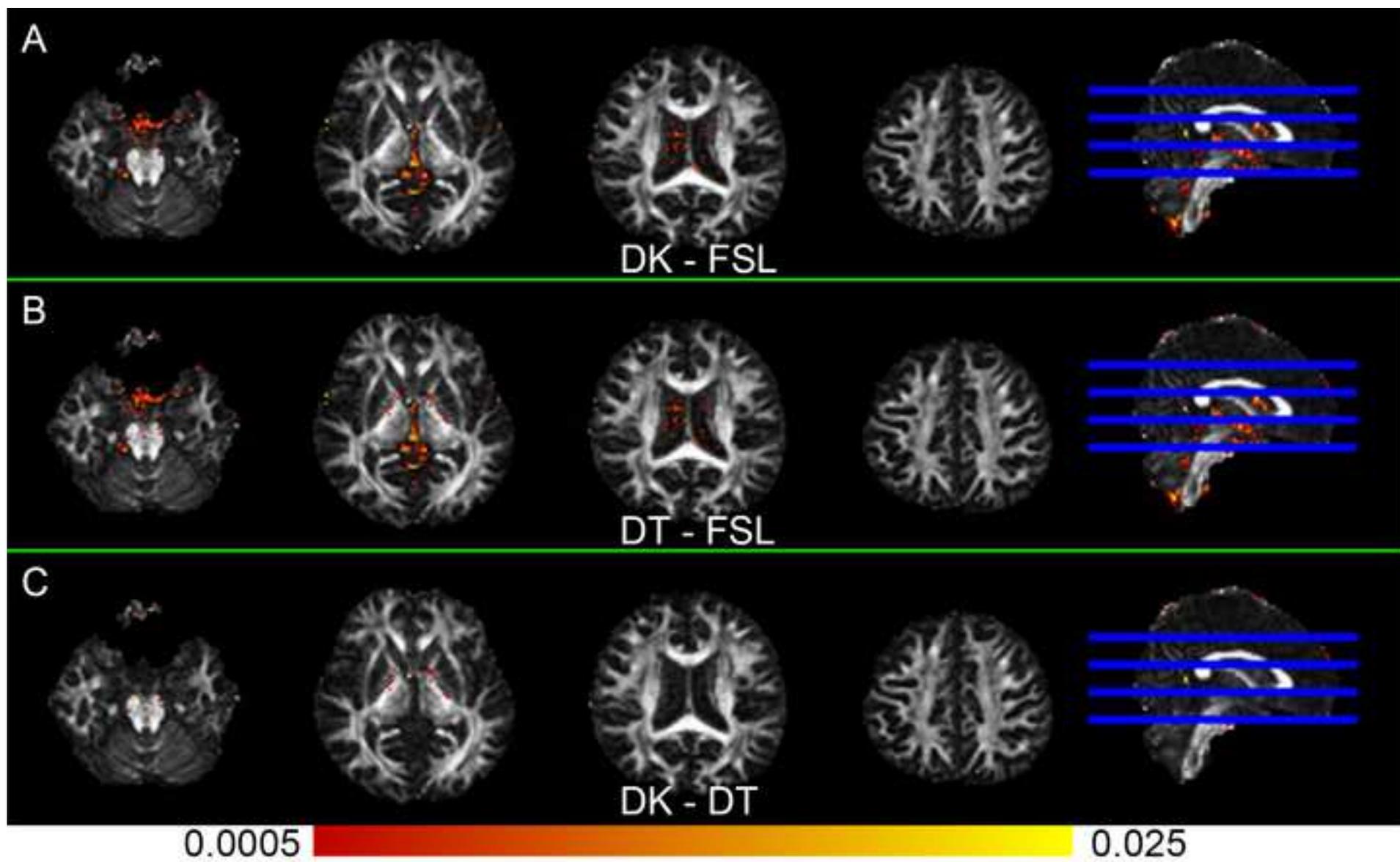


Figure8
[Click here to download high resolution image](#)

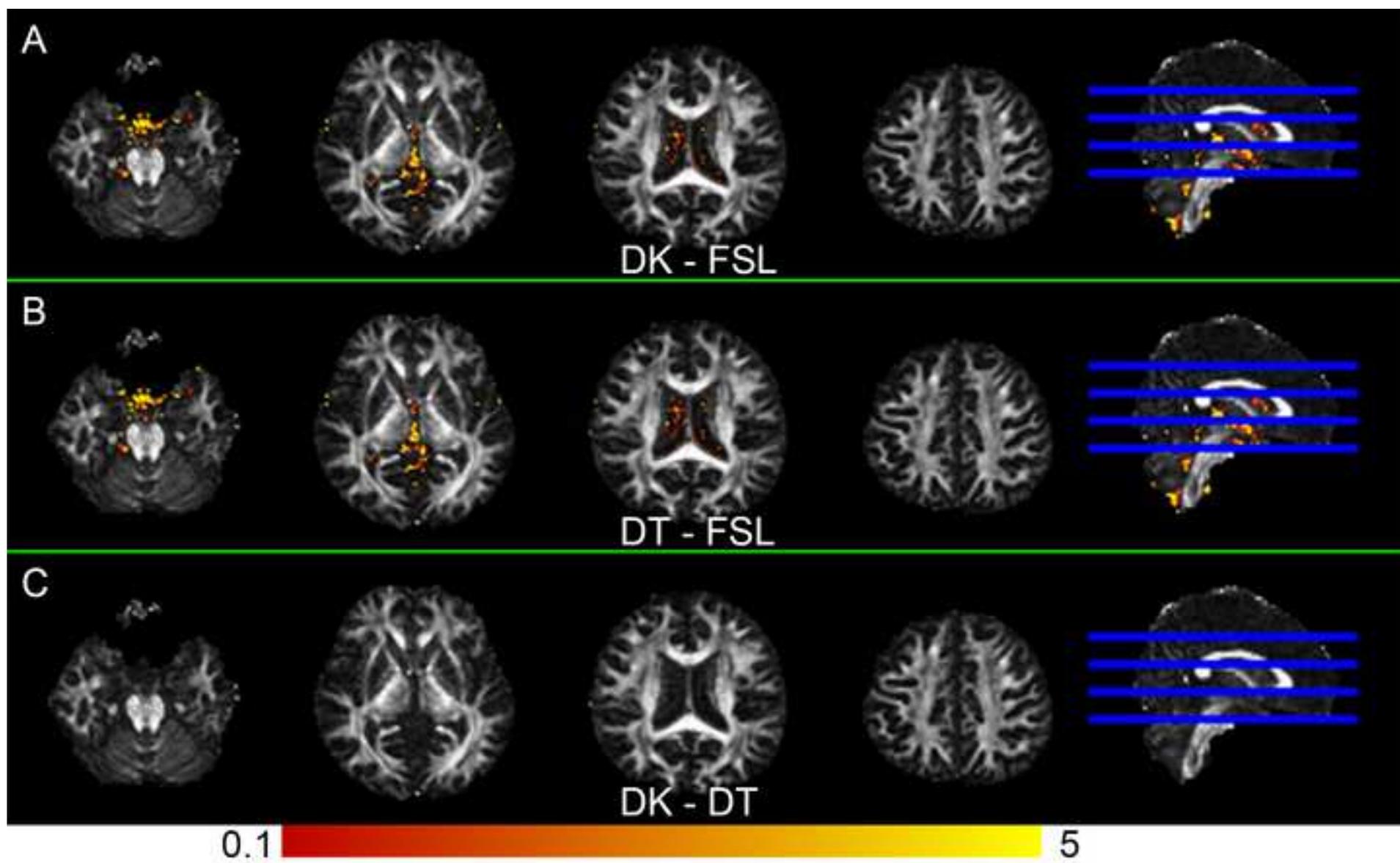


Figure9
[Click here to download high resolution image](#)

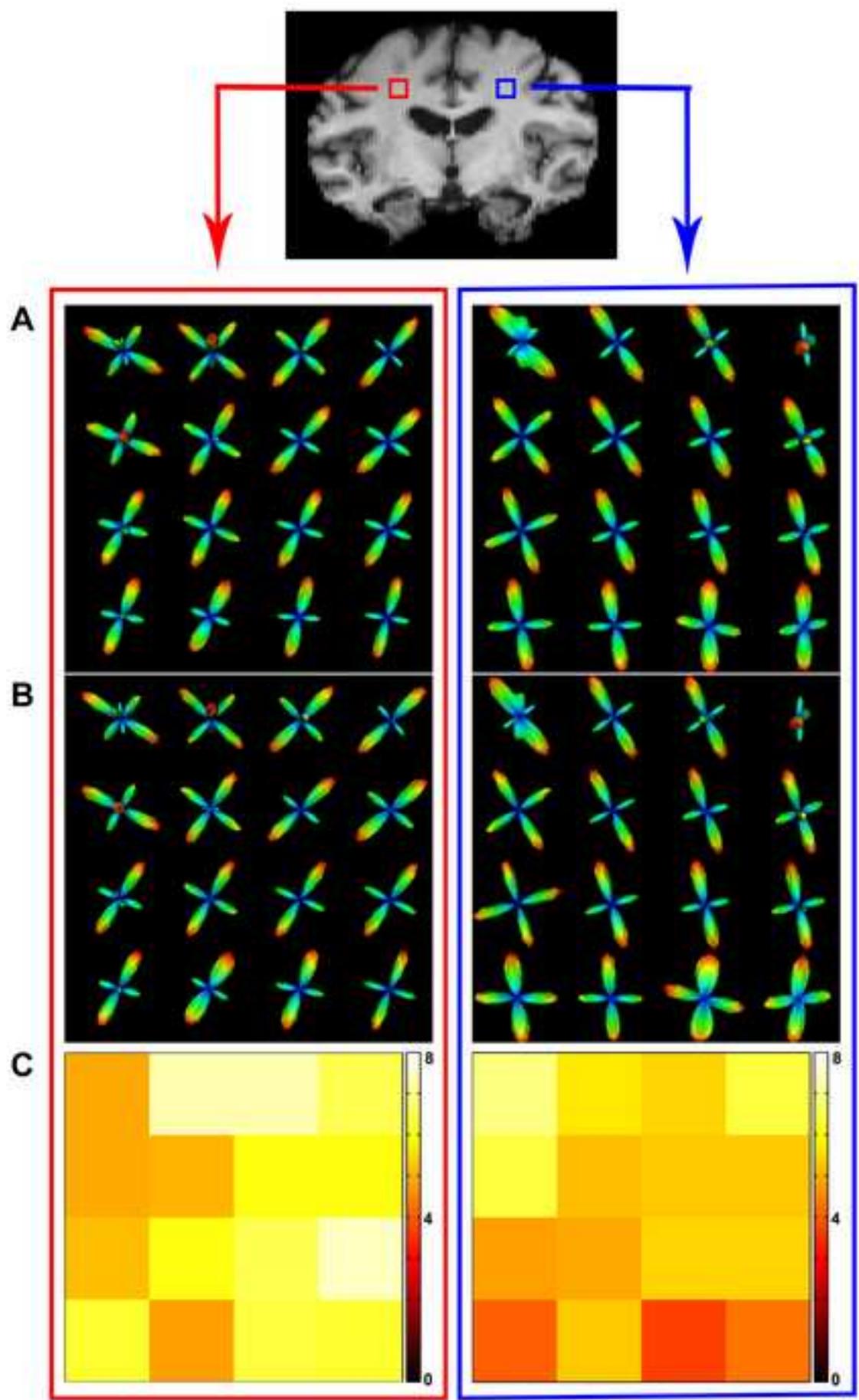


Figure10
[Click here to download high resolution image](#)

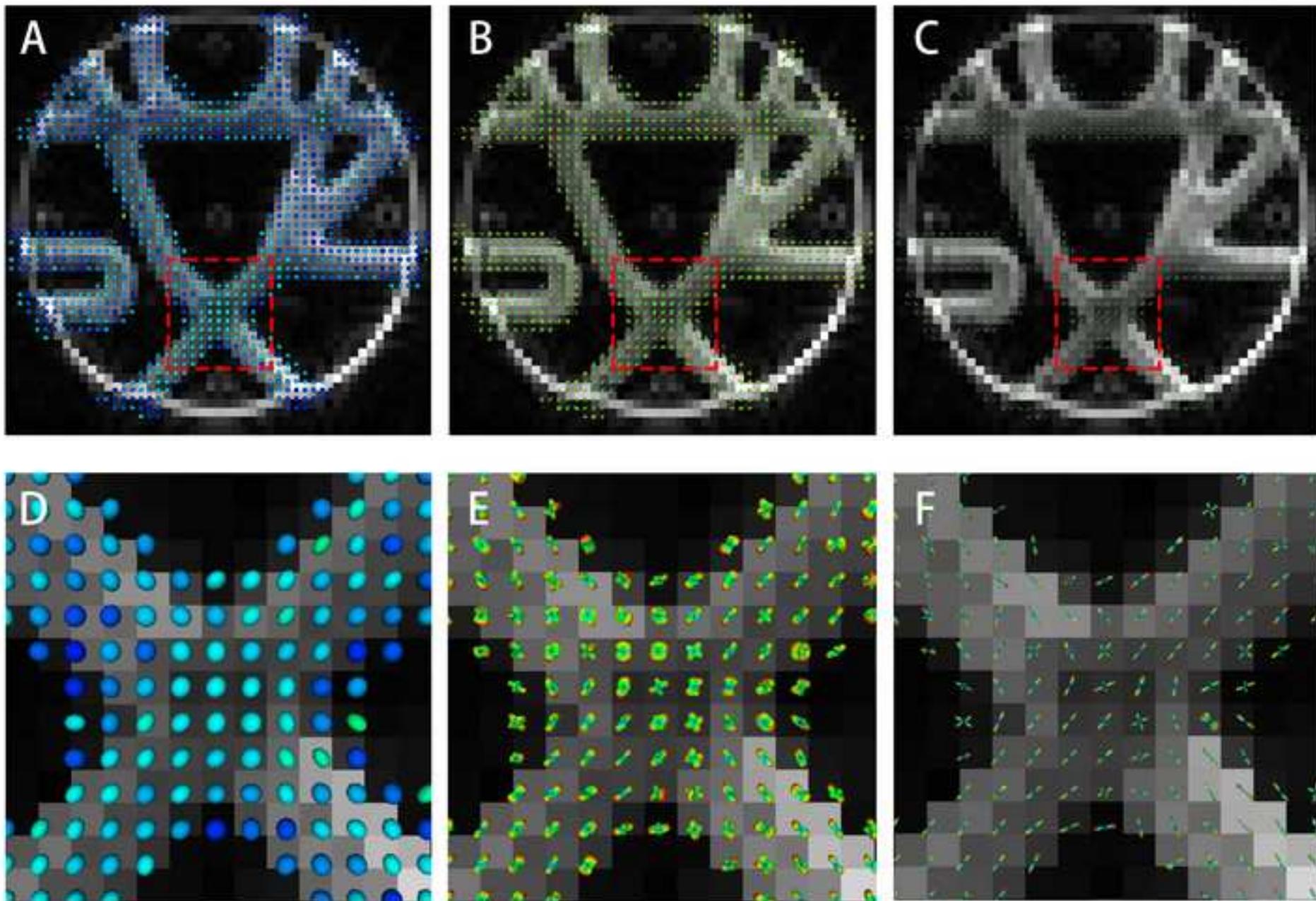
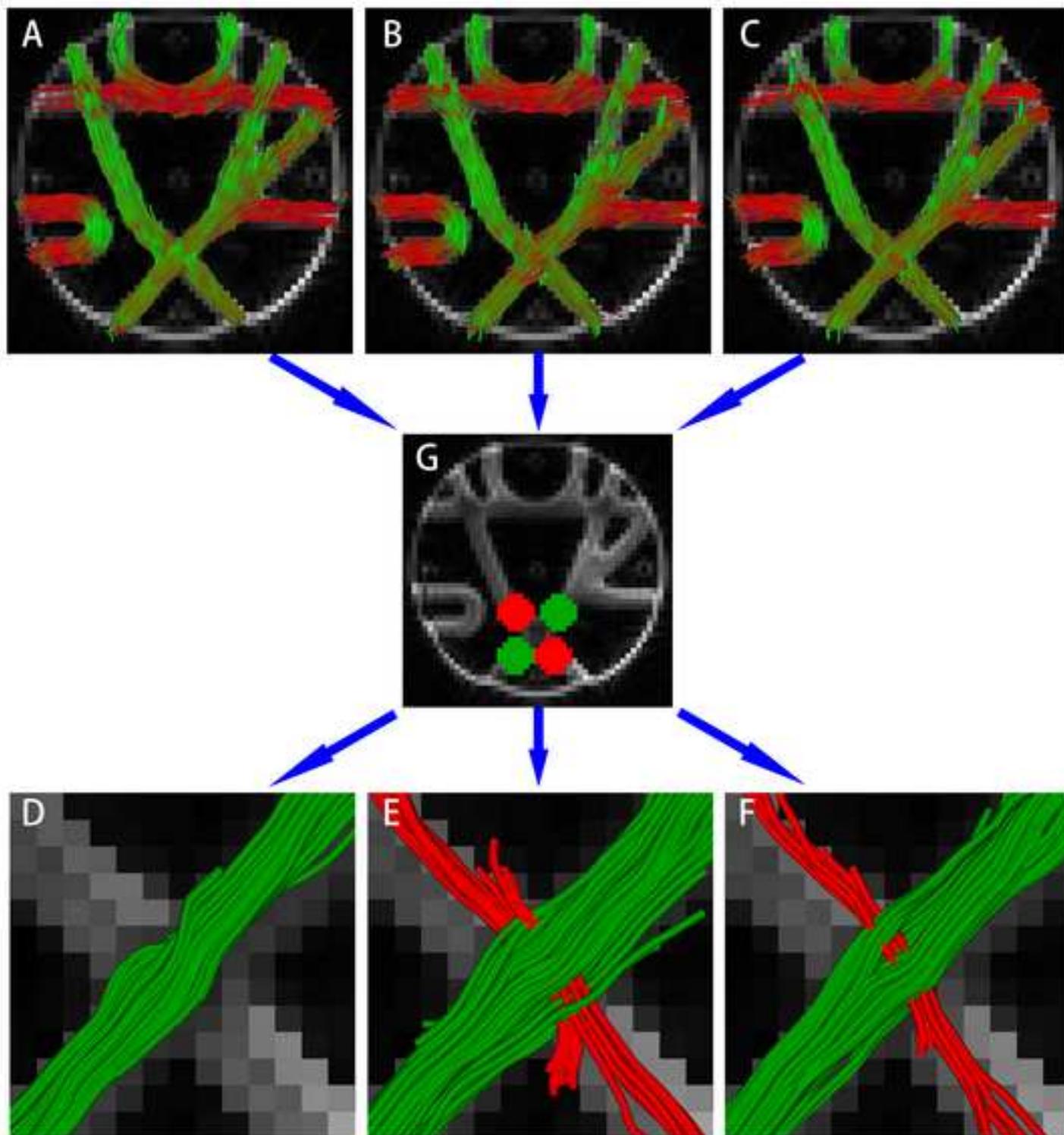


Figure11
[Click here to download high resolution image](#)



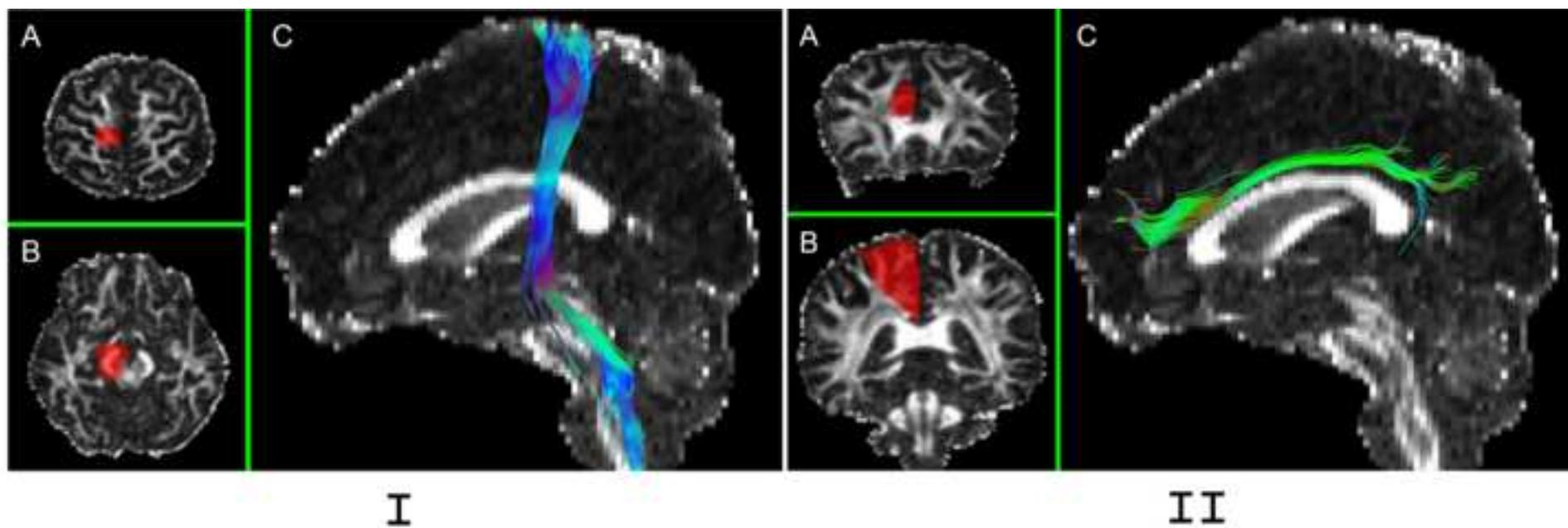


Figure13

[Click here to download high resolution image](#)

