

DKE Fiber Tractography Module: User's Guide

Version 1

Release Date: July 2015

Author:

G. Russell Glenn, B.S., B.A.
Medical Scientist Training Program (MD / PhD)

Department of Neurosciences
Department of Radiology and Radiological Science
Center for Biomedical Imaging
96 Jonathan Lucas St., MSC 323
Charleston, SC 29425-0323
Email: glennng@musc.edu



The DKE Fiber Tractography (FT) Module performs white matter FT using the DKI approximation of the diffusion orientation distribution function (dODF). This module uses the diffusion and kurtosis tensors from DKE and should be run with DKE version 2.6 or later. The following publications provide additional information regarding the DKI dODF:

1. Lazar et al. Estimation of the orientation distribution function from diffusional kurtosis imaging. *Magn Reson Med.* 2008;60:774-81.
2. Jensen et al. Leading non-Gaussian corrections for diffusion orientation diffusion orientation distribution function. *NMR Biomed.* 2014;27:202-11.
3. Neto Henriques et al. Exploring the 3D geometry of the diffusion kurtosis tensor-impact on the development of robust tractography procedures and novel biomarkers. *Neuroimage.* 2015;111:85-99.
4. Glenn et al. Optimization of white matter fiber tractography with diffusional kurtosis imaging. *NMR Biomed.* [In Press].

This module uses the closed form solution of the kurtosis dODF derived in Jensen et al. *NMR Biomed.* 2014;27:202-11 along with the image processing algorithms developed in Glenn et al. *NMR Biomed.* [In Press].

This software is compatible with:

DSI Studio (dsi-studio.labsolver.org, Department of Psychology, Carnegie Mellon University)
TrackVis (trackvis.org, Martinos Center for Biomedical Imaging, Massachusetts General Hospital)
Matlab (MathWorks, Natick, MA)

See OUTPUT section for more details about working with these programs.

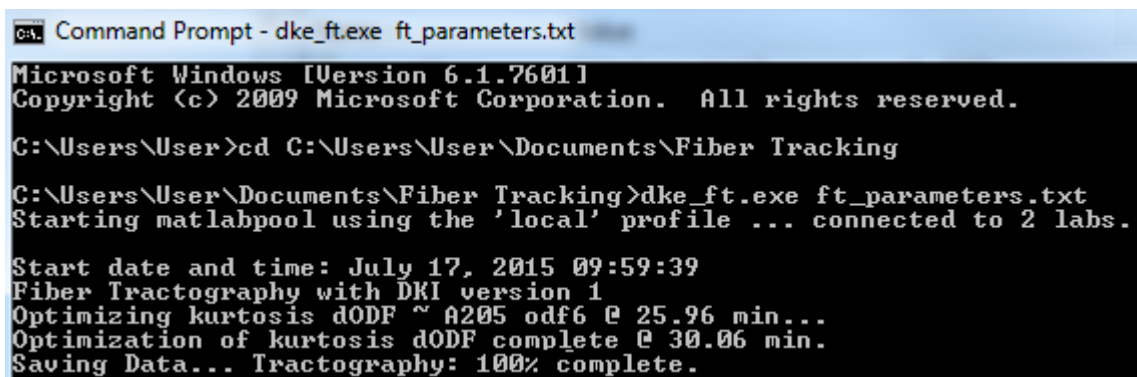
GETTING STARTED:

This software is available through NITRIC at <https://www.nitrc.org/projects/dke/> and requires the installation of the MATLAB Compiler Runtime 2012a (MCR version 7.17) available at: <http://www.mathworks.com/products/compiler/mcr/>. Once downloaded, the DKE FT Module will appear in your Downloads folder and can be moved to your Desktop or Program files.

The DKE FT Module is run using a file with MATLAB syntax. This file format is described below. In the download files you can find an example file called `ft_parameters.txt`.

To run the DKE FT Module launch a command window and execute the module along with `ft_parameters.txt` file. For example, in Windows, use the following command with user-specific pathways:

```
> C:/Users/Program Files/DKE/dke_ft.exe C :/Users/My Documents/Projects/ft_parameters.txt
```



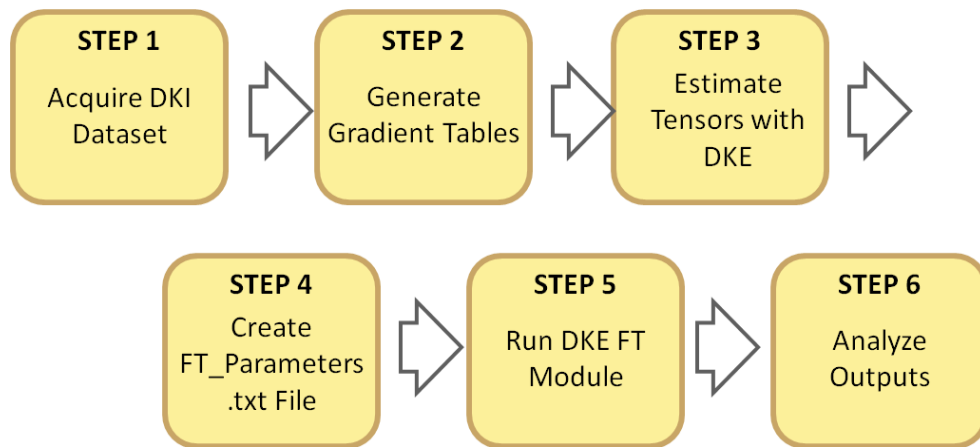
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Documents\Fiber Tracking

C:\Users\User\Documents\Fiber Tracking>dke_ft.exe ft_parameters.txt
Starting matlabpool using the 'local' profile ... connected to 2 labs.

Start date and time: July 17, 2015 09:59:39
Fiber Tractography with DKI version 1
Optimizing kurtosis dODF ~ A205 odf6 @ 25.96 min...
Optimization of kurtosis dODF complete @ 30.06 min.
Saving Data... Tractography: 100% complete.
```

STEPS:



NOTE: The DKE FT Module is designed to work with standard DKI datasets. However, in contrast to typical DKI parameters, such as mean kurtosis, the fiber bundle orientations used for tractography are *not* rotationally invariant. Therefore, generating a unique gradient table for each subject is required. We recommend using a program such as the `dcm2niigui` provided by MRICron (<https://www.nitrc.org/projects/mricron>) or an equivalent method. These tables can also be obtained directly from the Dicom header using the gradient directions along with the rotation matrix indicated by the `ImageOrientationPatient` field.

Since there is no standard way to interpolate the tensors, it is important to specify `map_interpolation_method.flag = 0` during tensor estimation with DKE. Alternatively, an `fa.nii` image can be calculated directly from the diffusion tensor, in which case, the Nifti header from the diffusion weighted images should be used as a template to ensure the voxel size, image dimension, and image orientation are correct.

USER INPUTS:

User inputs are specified in the `ft_parameters.txt` file. To modify the user inputs open the `ft_parameters.txt` file and make the desired changes prior to running the DKE FT Module. Each input parameter is described below:

`studydir` Path to the directory containing each subject within a study.
`subject_list` Matlab cell containing strings to identify sub-directories within `studydir`, which are specific to each subject within the study, eg `{'Subject01', 'Subject02', 'Subject03', ...}`.

Each subject directory should contain a `DT.mat`, `KT.mat`, and `fa.nii` file specific to that subject as calculated by DKE or an alternative tensor estimation routine. The format of these files is discussed in the INPUT FILE FORMATS section.

ODF Optimization

These options are used to analyze the kurtosis dODF and estimate fiber bundle orientations from DKI. Output parameters such as `odf_k.mat`, `gfa.nii`, `nfd.nii`, etc. are generated from optimization of the kurtosis dODF.

`odf_optimization` Whether (1) or not (0) to optimize the kurtosis dODF? This step is necessary for DKI tractography to run, however once the outputs are generated, it is not necessary to repeat the entire `odf_optimization` to make changes to the fiber tractography algorithm. By setting this option to 0 changes can be made to the Tractography section without recalculating the orientations from the kurtosis dODF.

sd Which sampling distribution to use? These values are restricted to 3, 4, or 5 and determine how the kurtosis dODF is initially sampled during optimization.

sd = 3: Sample one half of 3 fold quadrisection of the icosahedron or 8 fold tessellation of the icosahedron with $n = 321$ vertices and an average separation angle between each peak and its nearest neighbors at $8.64(\pm 0.56)$ degrees.

sd = 4: Sample one half of 4 fold quadrisection of the icosahedron or 16 fold tessellation of the icosahedron with $n = 1281$ vertices and an average separation angle between each peak and its nearest neighbors at $4.32(\pm 0.28)$ degrees (recommended).

sd = 5: Sample one half of 5 fold quadrisection of the icosahedron or 24 fold tessellation of the icosahedron with $n = 5121$ vertices and an average separation angle between each peak and its nearest neighbors at $2.16 (\pm 0.14)$ degrees.

NOTE: orientations identified with sd may be refined iteratively with the non-linear quasi-Newton method. More details regarding the sampling distributions and optimization algorithm can be found in Glenn et al. NMR Biomed. [In Press].

quasiNewton Whether (1) or not (0) to apply non-linear optimization using the quasi Newton method with the Broyden–Fletcher–Goldfarb–Shanno algorithm to increase accuracy of peak detection.

radial_weight Radial weighting of the kurtosis dODF. By default radial_weight = 4.

release_memory This option was created to overcome a potential memory issue. This program uses the parallel computing toolbox provided by Matlab to help minimize the image processing time of this module. If this program slowly increases its memory usage the following workaround may be used to periodically release memory from a cache created for each available worker. We have been in contact with MathWorks regarding this issue and believe the issue to be resolved.

0: Do not release memory while running this script (recommended)
 1: Release the memory after running each subject (conservative approach)
 2: Release the memory more frequently (if memory resources are limited)

Additional strategies to reduce the memory resources of this program are discussed below.

wrt_flg Whether (1) or not (0) to write outputs?

pre_name Name to prepend to the beginning of each output file name. If the subject name is also written, as with batch processing, the subject name comes before the pre_name specified here. This is discussed further in the OUTPUTS section.

post_name Name to append to the end of each output file name. This is discussed further in the OUTPUTS section.

Tractography

These options are used to perform white matter fiber tractography using the scripts provided by the DKE FT Module, utilizing the Euler Method. The parameters used for each subject are automatically stored in the FT_struct.mat file. The tractography scripts create .mat and .trk files which can be used to analyze and visualize the tracts as discussed below.

tractography_flg Whether (1) or not (0) to perform tractography using the scripts provided with this module? Alternatively, tractography can be performed with DSI Studio using the orientations detected with the kurtosis dODF using the output .fib file.

fa_threshold FA cutoff threshold to apply during tractography. This causes streamlines to terminate if they reach a voxel with an FA value below the fa_threshold.

angle_threshold Angle cutoff threshold to apply during tractography (in degrees). This causes streamlines to terminate if they encounter an orientation that deviates by more than the angle_threshold from their current trajectory.

trk_length Minimum track length (in mm) to include. Tracks that do not exceed this minimum length are discarded.

step_size Step size (in mm) used during the Euler algorithm. By default this value is 1 mm but supplying 0 will use one half of the average voxel dimension length.

trk_mask File name or path to white matter tracking mask (in nifty format). This mask causes streamlines to terminate if they exit the mask, ie if a streamline enters a voxel where the trk_mask is zero, the streamline will terminate.

This mask is applied in addition to the FA mask defined with the fa_threshold (via intersection). To use this mask exclusively, set the fa_threshold to 0. If the trk_mask path is empty, then it is ignored. This mask should be in the same orientation as the fa.nii image, as specified by the image_orientation field (see below).

For batch processing, we suggest saving a mask with a common name (eg 'brain_mask.nii') in each subject folder. If a full path is not supplied, the program will search within the subject folder using the specified file name.

seednum Number of seeds to generate within the white matter tracking mask. Seeds are generated randomly throughout the mask. For alternative seeding strategies, use DSI Studio.

shift The tractography algorithm used here specifies the middle of the first voxel at coordinate (1,1,1) (prior to scaling for voxel dimensions), ie the bottom of the image volume is at (0.5,0.5,0.5). However, TrackVis specifies the bottom of the image volume to be (0,0,0). The shift term, is used to shift the coordinates down to the (0,0,0) origin.

output_DTI_trks Perform tractography separately based on the diffusion tensor estimated from DKE. This uses the diffusion tensor generated from the DKI signal model and is not, strictly speaking, equivalent to DTI tractography.

`image_orientation` Orientation of the image volume, eg LPS, RAS, etc. This is a string composed of three characters with each one indicating the orientation of the respective dimensions of the image volume, ie the first character specifies the orientation of the first dimension, etc. Allowable characters include:

L or R: voxel indices increase to the Left or Right, respectively.

A or P: voxel indices increase in the Anterior or Posterior orientation, respectively.

S or I: voxel indices increase in the Superior or Inferior orientation, respectively.

By default DKE outputs images in LAS format. However, the image orientation should *always* be carefully inspected.

`odf_orientation` Orientation of the gradient table used to estimate the tensors, eg LPS, RAS, etc. This is a string composed of three characters with each one indicating the orientation of the respective columns of the gradient table used for tensor estimation, ie the first character specifies the orientation of the first column, etc. Allowable characters include:

L or R: voxel indices increase to the Left or Right, respectively.

A or P: voxel indices increase in the Anterior or Posterior orientation, respectively.

S or I: voxel indices increase in the Superior or Inferior orientation, respectively.

In our experience, the gradient tables derived from SPM (eg using the `dcm2niigui`) are in LAS format. The gradient tables can also be extracted using DSI Studio or directly from the dicom images (correcting for the `ImageOrientationPatient` rotation), in which case they will be in LPS. The image orientation should *always* be carefully inspected.

Make DSI Studio .fib file

These options are used to generate a .fib file which can be read by DSI Studio. DSI Studio provides a variety of tools for visualizing the dODFs and performing additional analyzes, including tractography and connectivity, which can now be performed directly from DKI data. To make the DSI Studio .fib file, the `odf_optimization` parameter must be set to 1.

To load the .fib file, open DSI Studio, select “STEP3: Fiber tracking” and then open the .fib file. For more information of DSI Studio, see their documentation (<http://dsi-studio.labsolver.org/>). The .fib file may also be loaded into Matlab by changing the file extension to .mat and using Matlab’s load / save commands. To make changes to the .fib file from Matlab specify ‘-v4’ during save.

DSI Studio uses quantitative anisotropy (QA) to filter spurious ODFs peaks. However, in the `dki.fib` file this parameter has been replaced by the FA value from DKI, applying a QA threshold with the `dki.fib` file in DSI Studio is equivalent to filtering based on FA.

`make_fib_file` Whether (1) or not (0) to make the .fib file from the kurtosis dODF.

`save_odfs` Whether (1) or not (0) to save odfs within the .fib file. This step is used to visualize the kurtosis dODFs in DSI Studio. DSI Studio does not need the dODF variables to run tractography, since the orientation estimates are supplied directly. Saving all of the dODFs within the .fib file may require a significant amount of memory. If you are only interested in tractography and have limited memory resources, this option may be turned off.

scale_odfs	By default the kurtosis dODFs are scaled to a maximum value of 1 (<i>but not min-max normalized</i>). This option allows you to choose the maximum scaled value of the dODFs. This only affects visualization. Choosing a lower value can provide more control over the dODF visualization.
odf_size	Size of the dODF variables saved within the .fib file, where each dODF variable is an $n \times$ odf_size array where n is the number of dODF vertices saved and odf_size is the value specified. The kurtosis dODF is analyzed in blocks the size of odf_size based on the way the .fib file is generated. Decreasing the odf_size requires less memory.
odf_res	'low' or 'high,' specifying the the number of vertices to save. Each odf variable within the .fib file is an $n \times$ odf_size array where n is the number of odf vertices saved. If memory is an issue, the odfs can be down sampled to 321 vertices using the 'low' resolution option. For smoother 3D odf renderings, the 'high' resolution option is recommended.

OUTPUTS:

The output file names from this program are modified using the following convention:

file name = subject_name + pre_name + output_name + post_name + file_extention

By default, the pre_name and post_name options are empty. However, these may be useful, for example, when performing multiple analyses on the same subject folder.

Output Files

DKI_FT.trk	DKI fiber tracts in .trk format to be viewed with TrackVis. These tracts are in LPS orientation with the bottom of the image volume corresponding to (0,0,0). This file can also be read and modified in Matlab using trk_read / trk_write functions provided by John Colby from the UCLA Developmental Cognitive Neuroimaging Group (Sowell Lab).
DKI_FT.mat	DKI fiber tracts in Matlab format. The .mat contains a cell with each element corresponding to the vector coordinates of an individual streamline in LPS coordinate space, with the bottom of the image volume corresponding to (0,0,0). This file can be read and modified in Matlab using the load / save commands.
DTI_FT.trk	DTI fiber tracts in .trk format to be viewed with TrackVis. These tracts are in LPS orientation with the bottom of the image volume corresponding to (0,0,0). This file can also be read and modified in Matlab using trk_read / trk_write functions provided by John Colby from the UCLA Developmental Cognitive Neuroimaging Group (Sowell Lab).
DTI_FT.mat	DTI fiber tracts in Matlab format. The .mat contains a cell with each element corresponding to the vector coordinates of an individual streamline in LPS coordinate space, with the bottom of the image volume corresponding to (0,0,0). This file can be read and modified in Matlab using the load / save commands.
dkf.fib	The .fib file format is created to utilize additional visualization and analysis tools provided by DSI Studio. This file contains orientation estimates from the kurtosis dODF as well as

kurtosis dODF values for visualization. DSI Studio can also be used to perform tractography with different masking options. See DSI Studio documentation (dsi-studio.labsolver.org/) for more info.

- gfa.nii** Generalized fractional anisotropy (GFA) map. This parameter evaluates anisotropy in the kurtosis dODF and is conceptually analogous to the diffusion tensor derived FA map with the added benefit of being able to accommodate more complex diffusion profiles. See Tuch. Magn Reson Med. 2004;52:1358-72 and Glenn et al. NMR Biomed. 2015;28:448-59 for more info.
- gfa_rgb.mat** GFA color map image stored as a 4D image with the 1st and 2nd dimensions representing the 1st and 2nd dimensions of the input image, the 3rd dimension representing the RGB color values, and the 4th dimension representing the 3rd dimension of the input image. The GFA color map image uses the vector $v = gfa * n$, where n is the orientation of the global maximum of the kurtosis dODF for a given voxel and gfa is the corresponding gfa value, as the RGB values in a color image. This format is used so the entire RGB image volume is saved in one file and if gfa and gfa_rgb images are loaded into matlab, slices can be viewed using the commands `imagesc(gfa(:, :, slice))` or `imagesc(gfa_rgb(:, :, :, slice))`. If the input images are provided in LAS format, then the slice variable is in the transverse plane. This image volume can be converted to a coronal orientation by `gfa_rgb_cor = flipdim(permute(gfa_rgb, [4 1 3 2]), 1)` and to a sagittal orientation by `gfa_rgb_sag = flipdim(permute(gfa_rgb, [4 2 3 1]), 1)`.
- fa_rgb.mat** FA color map image stored in the same format as the gfa_rgb image discussed above.
- nfd.nii** Number of fiber directions (NFD) quantifies the number of local maxima pairs detected in the kurtosis dODF, corresponding to the number of fiber bundle orientations estimated in each voxel.
- odf_k.mat** Matlab cell containing the fiber bundle orientations predicted from the kurtosis dODF in each voxel. Each cell is a $3 \times n$ vector where n is the NFD value for that voxel and each column is a fiber bundle orientation vector. Each input in the odf_k cell represents the corresponding voxel in from the input images, and the fiber bundle orientation vectors are in the same coordinate system specified by the $odf_orientation$ input. The vectors are normalized and sorted by their dODF magnitudes, where the first column represents the global maximum, the second represents the second largest local maxima, etc. These orientations are used for tractography and ultimately govern the trajectory of streamlines.
- odf_k_max.mat** Kurtosis dODF value corresponding to each vector in the $odf_k.mat$ cell, which represents the dODF value for each local maxima detected in the kurtosis dODF. The min / max dODF values can be used, for example, to filter the dODF peaks or in applying min-max normalization for visualization.
- odf_k_min.mat** Minimum dODF value found over the spherical grid. The min / max dODF values can be used, for example, to filter the dODF peaks or in applying min-max normalization for visualization.
- odf_d.mat** Matlab cell representing the global maxima of the dODF evaluated from the diffusion tensor. This is equivalent to the principal eigenvector of the diffusion tensor used for DTI-based tractography. The odf_d variable is equivalent to the odf_k variable for the diffusion tensor.

odf_coeff.mat Kurtosis dODF coefficients that can be used to reconstruct the kurtosis dODF for each voxel. These correspond to the coefficients defined in Glenn et al. NMR Biomed [In Press], where each column is given by $[A1, B11, B12, B13, B22, B23, B33, C1111, C1112, C1113, C1122, C1123, C1133, C1222, C1223, C1233, C1333, C2222, C2223, C2233, C2333, C3333, U11, U22, U33, U12, U13, U23, \alpha]^T$. These coefficients can be used with the function provided in the KURTOSIS dODF COEFFICIENT FUNCTION section, and can be useful, for example, when visualizing or analyzing the kurtosis dODFs in Matlab.

TrackDensity_DKI.nii Track density image from DKI tractography. Track density images are formed by counting the number of streamlines that pass through a given voxel and dividing by the voxel volume with units tracts / mm^3 .

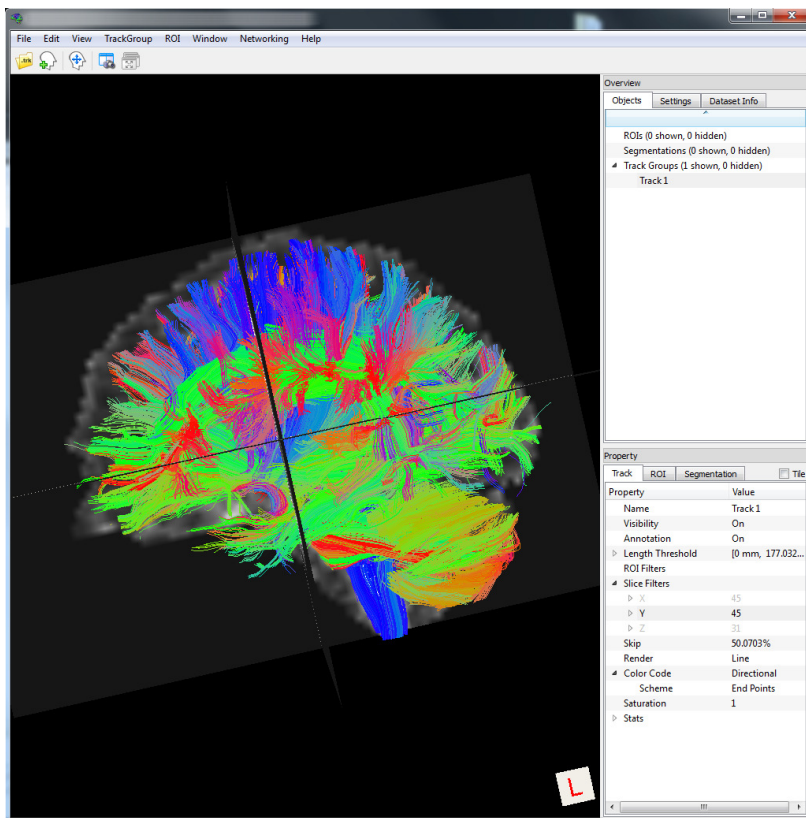
TrackDensity_DTI.nii Track density image from DTI tractography. Track density images are formed by counting the number of streamlines that pass through a given voxel and dividing by the voxel volume with units tracts / mm^3 .

FT_struc.mat Structure containing the parameters used for tractography including the user-defined tractography parameters, the image header and seed point distribution created.

Additional Programs

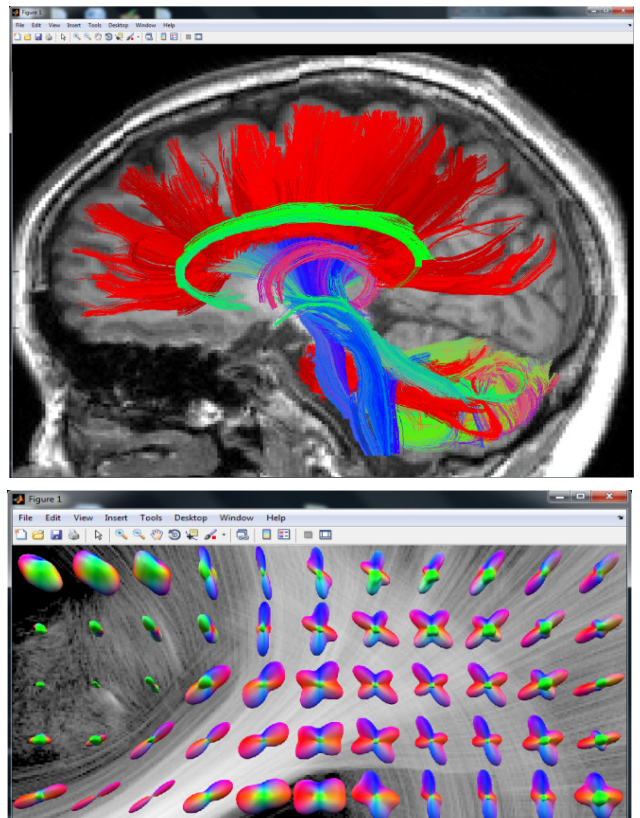
The DKI FT Module is designed to be compatible with additional programs, which may be used to visualize and analyze the output, including DSI Studio, TrackVis, and Matlab. Screen shots from each of these programs are given below. Please see the respective documentation for how to download and run each program.

TrackVis



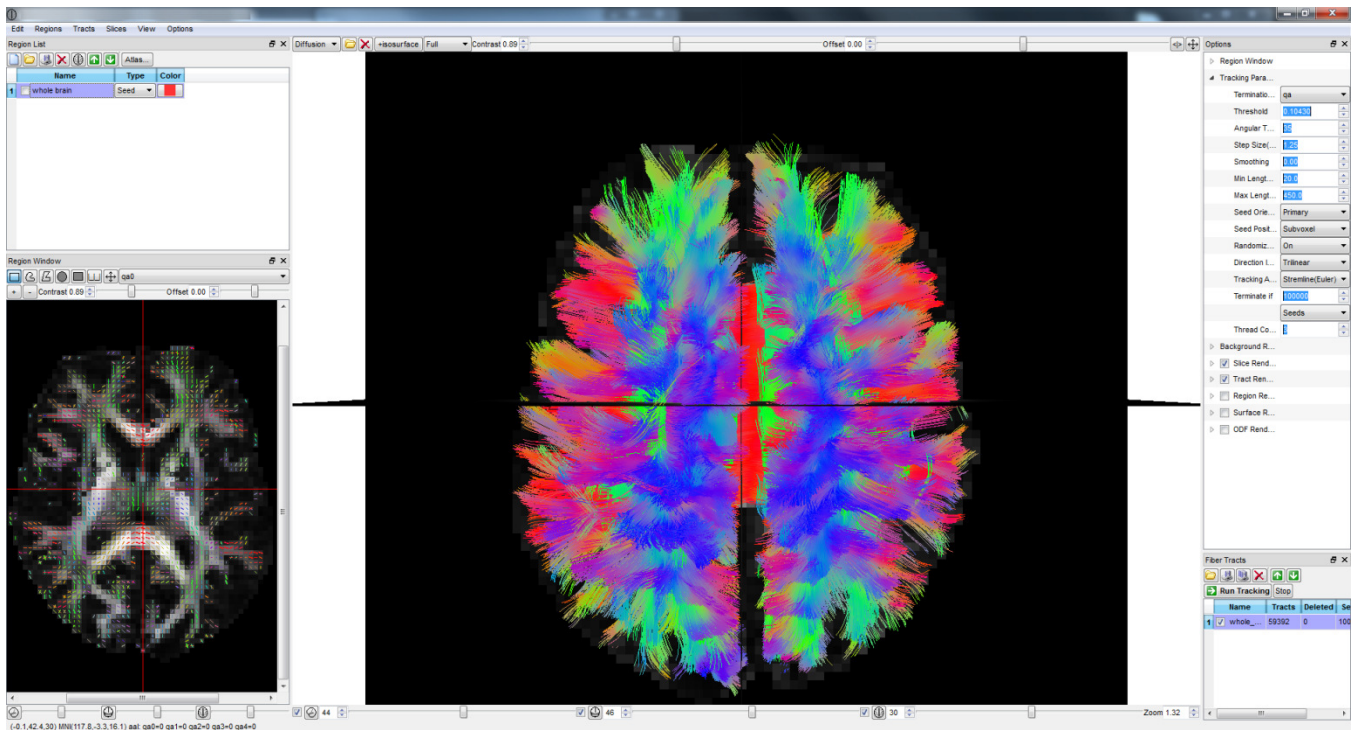
TrackVis is a visualization tool that can be used for 3D visualization of white matter fiber tracts and to group specific tracts with multiple ROI tools.

Matlab

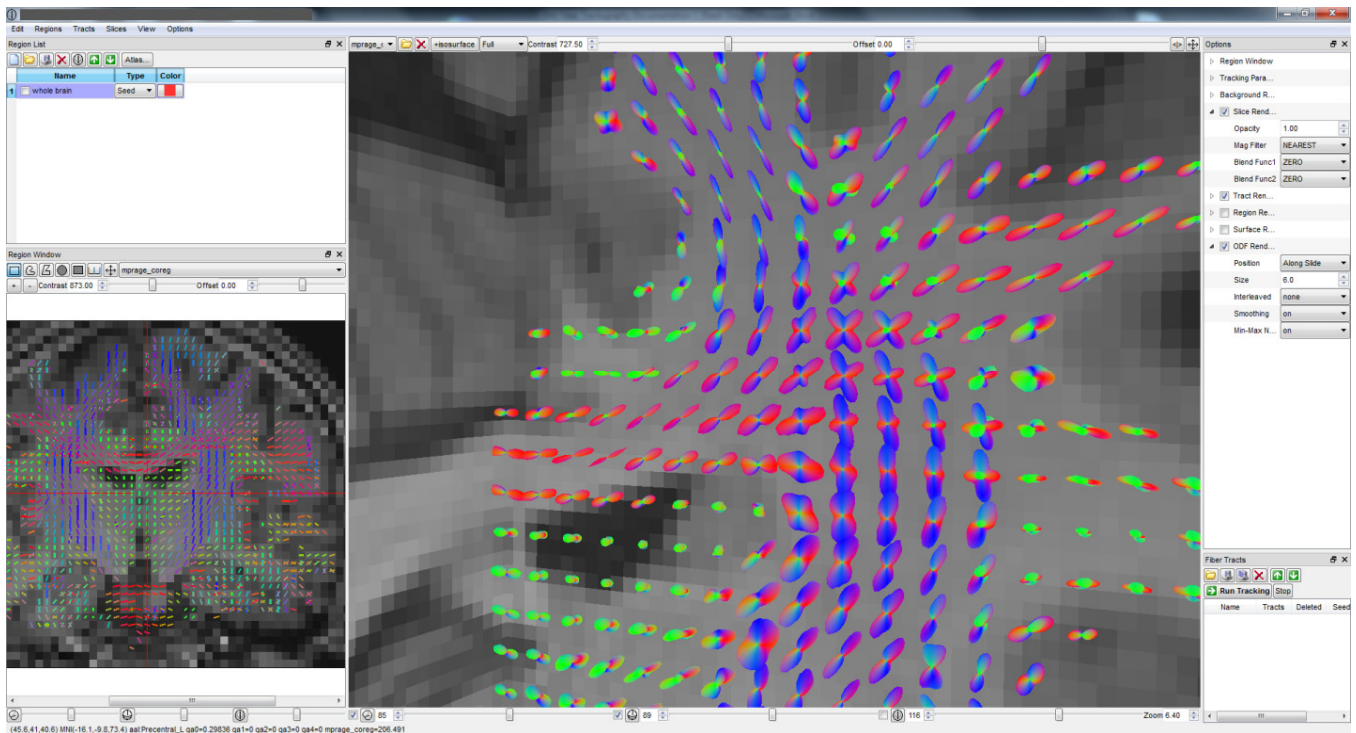


Matlab is a high-level programming language which can be used to develop custom analyses and visualization tools.

DSI Studio



DSI Studio is a versatile software package for performing analyses and for the 3D visualization of tractography data. DSI studio may be used for performing tractography from the kurtosis dODF values provided by the DKE FT Module, and can be useful, for example, when selecting specific seed point regions.



DSI Studio may also be used for the 3D visualization of the kurtosis dODFs computed by the DKE FT Module.

NOTE ON ORIENTATIONS:

It is important for this program to know both the orientation of the images and the gradient tables used to estimate the tensors. By default the output images from DKE will be stored in LAS orientation. We recommend using SPM to extract the gradient tables, and in our experience using SPM, they will also be in LAS orientation. However, other programs may use LPS or other orientations, eg the gradients stored in the Dicom header as well as gradients extracted by DSI studio are in LPS orientation. Since the gradient directions are not rotationally invariant it is important to know their precise orientation for tractography. The orientation of the image volumes and gradient tables is given in the `ft_parameters.txt` file used to run this program.

Outputs related to the kurtosis dODF, such as `odf_k.mat`, `gfa.nii`, `nfd.nii`, etc. are stored using the same orientation as the images and gradient table supplied. However, for the `.trk` and `.fib` files, the images and gradient tables are converted to LPS for compatibility with other programs.

NOTE FOR BATCH PROCESSING:

DKE assumes data is organized using the following file structure:

`studydir` directory containing each subject within a study
`subject_list` Matlab cell containing strings to identify sub-directories within `studydir` specific to each subject within the study, eg {'Subject01','Subject02','Subject03',...}.

In this case, `subject_list` is not empty, and DKE prepends the subject name to each output file in the corresponding subject directory.

To run one subject individually, `studydir` can specify the path to an individual subject's folder and the subject list may be specified as `subject_list = {}`. In this case the subject name will not be appended to the output file name.

If batch processing, as described above, was used by DKE, then the same `studydir`, `subject_list` convention should be used so that the FT Module can find the correct files.

INPUT FILE FORMATS:

The DKE FT Module is designed to work with the tensors estimated from DKE, which are output from DKE version 2.6 and later. The DKE FT Module requires three input data structures to run, which are discussed below. If tensor estimation is performed with a different program, the DKE FT Module can still be run, provided the data structures are supplied in the same format.

`DT.mat` 6-by-n vector containing unique elements of the diffusion tensor (mm²/s units), where n is the number of voxels within the image volume. The order that each column is indexed relates to the orientation of the image volumes used by DKE, typically LAS.

Each column is given by: $DT(:,i) = [D_{11}, D_{22}, D_{33}, D_{12}, D_{13}, D_{23}]^T$, where D is the diffusion tensor for the ith voxel.

`KT.mat` 15-by-n vector containing unique elements of the kurtosis tensor (mm²/s units), where n is the number of voxels within the image volume. The order that each column is indexed relates to the orientation of the image volumes used by DKE, typically LAS.

Each column is given by: = [W1111, W2222, W3333, W1112, W1113, W1222, W1333, W2223, W2333, W1122, W1133, W2233, W1123, W1223, W1233]^T, where W is the diffusion tensor for the ith voxel.

fa.nii NIFTI image volume for the fractional anisotropy (FA) parameter calculated from the diffusion tensor. The header data from this file is used as a template to generate the outputs from the DKE FT Module, including .nii and .trk file formats. This image may also be used as a cutoff threshold for tractography.

KURTOSIS dODF COEFFICIENT FUNCTION:

The DKE FT Module outputs a file called `odf_coeff.mat` where each column represents the coefficients for the kurtosis dODF for a given voxel in the image volume. To evaluate the kurtosis dODF, these coefficients can be supplied as inputs to the `kurtosis_dODF` function below, where `x` is an $n \times 2$ matrix of n spherical coordinates with the first column representing the azimuthal angle and the second column representing the polar angle for each coordinate, and `A` is the vector of dODF coefficients for a given voxel.

For more information, see Glenn et al. NMR Biomed. [In Press].

```
kurtosis_dODF =
@(x,A)(1./((sin(x(:,1)).*cos(x(:,2))).^2.*A(23)+(sin(x(:,1)).*sin(x(:,2))).^2.*A(24)+cos(x(:,1)).^2.*
A(25)+2.*(sin(x(:,1)).*cos(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))).*A(26)+2.*(sin(x(:,1)).*cos(x(:,2))).*cos(x(:,1)).*A(27)+
2.*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1)).*A(28))).^(A(29)+1)/2).*(1+(A(1)+(A(2).*(sin(x(:,1)).*cos(x(:,2))).^2+A(3).*
(sin(x(:,1)).*cos(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))))+A(4).*(sin(x(:,1)).*cos(x(:,2))).*cos(x(:,1))+A(5).*(sin(x(:,1)).*sin(
x(:,2))).^2+A(6).*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1))+A(7).*cos(x(:,1)).^2)/((sin(x(:,1)).*cos(x(:,2))).^2.*A(23)+(si
n(x(:,1)).*sin(x(:,2))).^2.*A(24)+cos(x(:,1)).^2.*A(25)+2.*(sin(x(:,1)).*cos(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))).*A(26)+
2.*(sin(x(:,1)).*cos(x(:,2))).*cos(x(:,1)).*A(27)+2.*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1)).*A(28))+A(8).*(sin(x(:,1)).*c
os(x(:,2))).^4+A(9).*(sin(x(:,1)).*cos(x(:,2))).^3.*(sin(x(:,1)).*sin(x(:,2))))+A(10).*(sin(x(:,1)).*cos(x(:,2))).^3.*cos(x(
(:,1))+A(11).*(sin(x(:,1)).*cos(x(:,2))).^2.*(sin(x(:,1)).*sin(x(:,2))).^2+A(12).*(sin(x(:,1)).*cos(x(:,2))).^2.*(sin(x(:,1))
.*sin(x(:,2))).*cos(x(:,1))+A(13).*(sin(x(:,1)).*cos(x(:,2))).^2.*cos(x(:,1)).^2+A(14).*(sin(x(:,1)).*cos(x(:,2))).*(sin(x
(:,1)).*sin(x(:,2))).^3+A(15).*(sin(x(:,1)).*cos(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))).^2.*cos(x(:,1))+A(16).*(sin(x(:,1)).*c
os(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1)).^2+A(17).*(sin(x(:,1)).*cos(x(:,2))).*cos(x(:,1)).^3+A(18).*(sin(x(:,1)
)).*sin(x(:,2))).^4+A(19).*(sin(x(:,1)).*sin(x(:,2))).^3.*cos(x(:,1))+A(20).*(sin(x(:,1)).*sin(x(:,2))).^2.*cos(x(:,1)).^2+
A(21).*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1)).^3+A(22).*cos(x(:,1)).^4)/((sin(x(:,1)).*cos(x(:,2))).^2.*A(23)+(sin(x(:,1)
)).*sin(x(:,2))).^2.*A(24)+cos(x(:,1)).^2.*A(25)+2.*(sin(x(:,1)).*cos(x(:,2))).*(sin(x(:,1)).*sin(x(:,2))).*A(26)+2.*(sin(
x(:,1)).*cos(x(:,2))).*cos(x(:,1)).*A(27)+2.*(sin(x(:,1)).*sin(x(:,2))).*cos(x(:,1)).*A(28)).^2)/24);
```